# BIOS LISTING
# for
# CP/M 2.2.03

595-2775

```
0000                ORG     000H
0001 =      H17T    EQU     1
0000 =      H37T    EQU     0
0000 =      H47T    EQU     0
0000 =      H67T    EQU     0

            ;+*******************
            ;+
            ;+  * * * N O T E * * *
            ;+
            ;+  THE ABOVE 5 LINES OF CODE ARE THE PREAMBLE TO THE BIOS.
            ;+  THESE LINES ARE USED BY 'MAKEBIOS' IN GENERATING THE BIOS.
            ;+  THESE LINES SHOULD NOT BE ALTERED FOR ANY REASON UNLESS THE
            ;+  PROGRAM 'MAKEBIOS' IS ALSO ALTERED.  THESE LINES MUST APPEAR
            ;+  AS THE FIRST 5 LINES IN THIS SOURCE.
            ;+
            ;+*******************

                PAGE
```

```
                        TITLE   'HEATH/ZENITH BIOS                    28 AUG 81'
0003 =          VERS    EQU     03
0020 =          LEVEL   EQU     ' '
0008 =          MONTH   EQU     08
001C =          DAY     EQU     28
0051 =          YEAR    EQU     81

                ;+*******************
                ;+
                ;+  BIOS2, A BIOS MODULE FOR CP/M 2.2
                ;+  FOR USE WITH HEATH/ZENITH H/Z89 AND H-8 COMPUTERS
                ;+      AND H17/H77/H87 5 1/4 INCH DISKS
                ;+      AND H47/Z47     8     INCH DISKS
                ;+      AND H37         5 1/4 INCH DISKS
                ;+      AND H67        HARD DISK WITH 8 INCH FLOPPY
                ;+
                ;+ COPYRIGHT 1980,1981 HEATH COMPANY, BENTON HARBOR, MICHIGAN
                ;+
                ;*      JIM TITTSLER / BILL ZURNEY
                ;+      HEATH/ZENITH SOFTWARE GROUP
                ;+      HILLTOP ROAD
                ;+      SAINT JOSEPH, MICHIGAN
                ;
                ;  BECAUSE THIS CODE MUST BE ASSEMBLEABLE UNDER BOTH 'ASM' AND 'MAC',
                ;  THE CONDITIONALS MAY SEEM A LITTLE STRANGE.  'ASM' DOESN'T HAVE
                ;  THE FOLLOWING
                ;    1) IF/THEN/ELSE STRUCTURE
                ;    2) NESTED IF CAPABILITY
                ;    3) RELATIONAL OPERATORS
                ;+
                ;+*******************

0000 =          FALSE   EQU     0
0001 =          TRUE    EQU     1
                        IF      TRUE-1
                %:      TRUE NE 1
                        ENDIF

                        IF      (H17T+H37T+H47T+H67T-1) SHR 15
                %:      NO DISK DRIVE TYPES SPECIFIED
                        ENDIF
                        IF      (2-(H17T+H37T+H47T+H67T)) SHR 15
                %:      TOO MANY DISK DRIVE TYPES SPECIFIED
                        ENDIF

0000 =          PARTITN EQU     TRUE AND H67T   ;FALSE = NO HARD DISK PARTITION SUPPORT
                                                ;TRUE  = SUPPORT HARD DISK PARTITIONING

0000 =          H67PART2 EQU    TRUE AND PARTITN ;FALSE = ONLY 1 PARTITION AT A TIME
                                                 ;TRUE  = ALLOW 2 DRIVES (PARTITIONS)

0000 =          TOD     EQU     FALSE           ;TIME OF DAY HANDLER
0000 =          EVENT   EQU     FALSE           ;EVENT DOWN COUNTER
0001 =          INTINP  EQU     TRUE            ;ASSEMBLE WITH INTERRUPT CRT DRIVER
```

```
0000 =          BRKKEY  EQU     FALSE AND INTINP ;TRUE  = WARM START ON BREAK KEY
                                                 ;FALSE = NO SPECIAL PROCESSING
0000 =          H37ED   EQU     TRUE AND H37T    ;TRUE = BIOS SUPPORTS H37 EXTENDED
                                                 ;       DOUBLE DENSITY
0000 =          H47ED   EQU     TRUE AND H47T    ;TRUE = BIOS SUPPORTS H47 EXTENDED
                                                 ;       DOUBLE DENSITY

0003 =          H17ND   EQU     3*H17T           ;NUMBER OF H17 DRIVES SUPPORTED
0000 =          H37ND   EQU     3*H37T           ;NUMBER OF H37 DRIVES SUPPORTED
0000 =          H47ND   EQU     2*H47T           ;NUMBER OF H47 DRIVES SUPPORTED
0000 =          H67ND   EQU     2*H67T+H67PART2  ;NUMBER OF H67 DRIVES SUPPORTED

0000 =          BIOS    EQU     $

F200 =          BDOS    EQU     BIOS-0E00H
EA00 =          CCP     EQU     BDOS-0800H
EA03 =          CCPCLR  EQU     CCP+3
0000 =          BOOT    EQU     0000H            ;BASE OF USABLE RAM
0003 =          IOBYTE  EQU     BOOT+3           ;I/O DEVICE ASSIGNMENT BYTE
0004 =          LOGDSK  EQU     BOOT+4           ;WHERE CPM STORES DEFAULT DRIVE
0040 =          BDMAP   EQU     0040H            ;LOGICAL TO PHYSICAL DRIVE MAP
                                                 ; (UP TO 8 BYTES [DRIVES] 0040H-0047H)
0048 =          BBDF    EQU     0048H            ;BOOT DEVICE FLAGS
0049 =          BBDA    EQU     0049H            ;BOOT DEVICE ADDRESS (BASE PORT #)
004A =          BBP     EQU     004AH            ;BOOT BEGINNING OF PARTITION
                                                 ; (3 BYTE VALUE)
004D =          BUPB    EQU     004DH            ;BOOT LAST SECTOR OF PARTITION + 1
                                                 ; (3 BYTE VALUE)
004E =          BBIOS   EQU     004EH            ;CONTAINS ADDRESS OF START OF BIOS
                                                 ; AFTER COLD BOOT IS COMPLETED
005C =          FCB     EQU     BOOT+5CH         ;DEFAULT FILE CONTROL BLOCK
0080 =          BUFF    EQU     BOOT+80H         ;DEFAULT DISK BUFFER
0100 =          TPA     EQU     BOOT+100H        ;BASE OF TRANSIENT PGM AREA

                        PAGE
```

```
                        ;
                        ; H17 DISK RELATED EQUATES
                        ;

007C =          UPDP    EQU     07CH            ;DISK DATA PORT
007D =          UPFC    EQU     07DH            ;FILL CHARACTER
007D =          UPST    EQU     07DH            ;STATUS FLAGS
007E =          UPSC    EQU     07EH            ;SYNC CHARACTER (OUTPUT)
007E =          UPSR    EQU     07EH            ;SYNC RESET (INPUT)
007F =          DPDC    EQU     07FH            ;DISK CONTROL PORT

0002 =          U0      EQU     02H             ;H17    UNIT 0
0004 =          U1      EQU     04H             ;       UNIT 1
0008 =          U2      EQU     08H             ;       UNIT 2
0010 =          DFMO    EQU     10H             ;MOTOR ON (ALL DRIVES)
0020 =          DFDI    EQU     20H             ;DIRECTION (0 = OUT)
0040 =          DFST    EQU     40H             ;STEP COMMAND (ACTIVE HIGH)

0001 =          DFHD    EQU     01H             ;HOLE DETECT
0002 =          DFT0    EQU     02H             ;TRACK 0 DETECT
0004 =          DFWP    EQU     04H             ;WRITE PROTECT
0008 =          DFSD    EQU     08H             ;SYNC DETECT

00FD =          DSYN    EQU     0FDH            ;PREFIX SYNC CHARACTER

0014 =          LPSA    EQU     20              ;NUMBER OF TRIES FOR CORRECT SECTOR
0005 =          STSA    EQU     8/2+1           ;MS/2 TO WAIT FOR INDEX HOLE
0007 =          STSB    EQU     12/2+1          ;MS/2 TO WAIT PAST INDEX HOLE
0014 =          WHDA    EQU     20              ;UDLY COUNT FOR HOLE DEBOUNCE
0014 =          WHNA    EQU     20              ;UDLY COUNT FOR HOLE DEBOUNCE
0050 =          WSCA    EQU     64*25/20        ;LOOP COUNT FOR 25 CHARACTERS
0014 =          WRITA   EQU     20              ;GUARDBAND COUNT FOR WRITE
000A =          WRITB   EQU     10              ;NUMBER OF ZERO CHARACTERS AFTER HOLE EDGE
0010 =          WRITC   EQU     128/8           ;TWO CHARACTER DELAY BEFORE WRITING
0030 =          READA   EQU     48              ;DELAY BEFORE HUNT MODE
00FA =          SPD     EQU     250             ;250 * 4MS = 1 S
0014 =          HLTG    EQU     20              ; 20 * 4MS = 80 MS
0006 =          HST     EQU     24/4            ;HEAD SETTLE TIME 24 MS
000F =          STEPR   EQU     30/2            ;STEP RATE MS/2
060F =          DELAYS  EQU     6*256+15        ;HEAD LOAD AND MOTOR ON TIMER VALUES
000A =          RETRIES EQU     10              ;NUMBER OF RETRIES

0001 =          D$E$TRK EQU     001H            ;BAD TRACK ERROR
0002 =          D$E$HSY EQU     002H            ;HEADER SYNC ERROR
0004 =          D$E$HCK EQU     004H            ;HEADER CHECKSUM
0008 =          D$E$CHK EQU     008H            ;CHECKSUM ERROR
0010 =          D$E$RNF EQU     010H            ;RECORD NOT FOUND
0020 =          D$E$MDS EQU     020H            ;MISSING DATA SYNC
0040 =          D$E$WRP EQU     040H            ;WRITE PROTECT ERROR
0080 =          D$E$UNR EQU     080H            ;UNIT NOT READY
```

```
                        ;
                        ;  H47 DISK EQUATES.
                        ;
0000 =          H47CTL  EQU     0               ;STATUS/CONTROL PORT DISPLACEMENT
0001 =          H47DAT  EQU     1               ;DATA PORT DISPLACEMENT

                        ;  STATUS PORT BITS
0080 =          DSTR    EQU     10000000B       ;TR
0040 =          DSIE    EQU     01000000B       ;INT ENABLE
0020 =          DSDONE  EQU     00100000B       ;DONE (I.E. NOT BUSY)
0001 =          DSERR   EQU     00000001B       ;ERROR

                        ;  CONTROL PORT BITS
0040 =          DCIE    EQU     01000000B       ;INT ENABLE
0002 =          DCRES   EQU     00000010B       ;RESET

                        ;  COMMANDS
0001 =          DRS     EQU     01H             ;READ STATUS
0002 =          DRAS    EQU     02H             ;READ AUXILIARY STATUS
0003 =          DSNS    EQU     03H             ;SET NUMBER OF SECTORS
0007 =          DRD     EQU     07H             ;READ (BUFFERED)
0008 =          DWR     EQU     08H             ;WRITE (BUFFERED)
000B =          DCOPY   EQU     0BH             ;COPY
000D =          DFMT    EQU     0DH             ;SINGLE DENSITY FORMAT
000E =          DFMTD   EQU     0EH             ;FORMAT DOUBLE DENSITY (TRK 0 SINGLE)
000F =          DFMTD2  EQU     0FH             ;FORMAT DOUBLE DENSITY (TRK 0-76)

                        PAGE
```

```
                        ;
                        ;  H37 EQUATES
                        ;

                        ;  PORT ASSIGNMENTS
0078 =                  FD$BASE EQU     078H            ;BASE PORT ADDRESS
0078 =                  FD$CON  EQU     FD$BASE         ;DISK CONTROL PORT
0079 =                  FD$INT  EQU     FD$BASE+1       ;INTERFACE MUX PORT
007A =                  FD$CMD  EQU     FD$BASE+2       ;1797 COMMAND REGISTER
007A =                  FD$STA  EQU     FD$BASE+2       ;     STATUS REGISTER
007B =                  FD$DAT  EQU     FD$BASE+3       ;     DATA REGISTER
007A =                  FD$SEC  EQU     FD$BASE+2       ;     SECTOR REGISTER
007B =                  FD$TRK  EQU     FD$BASE+3       ;     TRACK REGISTER

                        ;  INTERFACE MUX PORT FLAGS
0000 =                  FD$CD   EQU     0               ;ACCESS C/D REGISTERS
0001 =                  FD$TS   EQU     1               ;ACCESS T/S REGISTERS

                        ;  COMMANDS
0000 =                  FDCRST  EQU     000H            ;RESTORE
0010 =                  FDCSEK  EQU     010H            ;SEEK
0020 =                  FDCSTP  EQU     020H            ;STEP
0040 =                  FDCSTI  EQU     040H            ;STEP IN
0060 =                  FDCSTO  EQU     060H            ;STEP OUT
0080 =                  FDCRDS  EQU     080H            ;READ SECTOR
00A0 =                  FDCWRS  EQU     0A0H            ;WRITE SECTOR
00C0 =                  FDCRDA  EQU     0C0H            ;READ ADDRESS
00E0 =                  FDCRDT  EQU     0E0H            ;READ TRACK
00F0 =                  FDCWRT  EQU     0F0H            ;WRITE TRACK
00D0 =                  FDCFI   EQU     0D0H            ;FORCE INTERRUPT

                        ;  TYPE 1 COMMAND FLAGS
0010 =                  FDFUTR  EQU     00010000B       ;UPDATE TRACK REGISTER
0008 =                  FDFHLB  EQU     00001000B       ;HEAD LOAD AT BEGINNING
0004 =                  FDFVRF  EQU     00000100B       ;VERIFY FLAGS

                        ;  TYPE 1 COMMAND STEP RATE FLAGS
0000 =                  FDFS6   EQU     00000000B       ;STEP RATE  6 MS
0001 =                  FDFS12  EQU     00000001B       ;           12
0002 =                  FDFS20  EQU     00000010B       ;           20
0003 =                  FDFS30  EQU     00000011B       ;           30

                        ;  TYPE 2&3 COMMAND FLAGS
0010 =                  FDFMRF  EQU     00010000B       ;MULTIPLE RECORD FLAG
0008 =                  FDFSLF  EQU     00001000B       ;SECTOR LENGTH FLAG
0004 =                  FDFDLF  EQU     00000100B       ;30 MS DELAY
0002 =                  FDFSS1  EQU     00000010B       ;SELECT SIDE 1
0001 =                  FDFDDM  EQU     00000001B       ;DELETED DATA MARK

                        ;  TYPE 4 COMMAND FLAGS
0000 =                  FDFINI  EQU     00000000B       ;TERMINATE WITH NO INTERRUPT
0001 =                  FDFII0  EQU     00000001B       ;NOT READY TO READY TRANSITION
0002 =                  FDFII1  EQU     00000010B       ;READY TO NOT READY TRANSITION
0004 =                  FDFII2  EQU     00000100B       ;INDEX PULSE
0008 =                  FDFII3  EQU     00001000B       ;IMMEDIATE INTERRUPT
```

```
                ;  STATUS FLAGS
0080 =          FDSNRD  EQU     10000000B       ;NOT READY
0040 =          FDSWPV  EQU     01000000B       ;WRITE PROTECT VIOLATION
0020 =          FDSHLD  EQU     00100000B       ;HEAD IS LOADED
0020 =          FDSRTE  EQU     00100000B       ;RECORD TYPE
0020 =          FDSWTF  EQU     00100000B       ;WRITE FAULT
0010 =          FDSSEK  EQU     00010000B       ;SEEK ERROR
0010 =          FDSRNF  EQU     00010000B       ;RECORD NOT FOUND
0008 =          FDSCRC  EQU     00001000B       ;CRC ERROR
0004 =          FDSTK0  EQU     00000100B       ;FOUND TRACK 0
0004 =          FDSLDT  EQU     00000100B       ;LOST DATA
0002 =          FDSIND  EQU     00000010B       ;INDEX HOLE
0001 =          FDSBSY  EQU     00000001B       ;BUSY

                ;  INFO RETURNED BY A READ ADDRESS COMMAND
0000 =          FDRATRK EQU     0               ;TRACK
0001 =          FDRASID EQU     1               ;SIDE
0002 =          FDRASEC EQU     2               ;SECTOR
0003 =          FDRASL  EQU     3               ;SECTOR LENGTH
0004 =          FDRACRC EQU     4               ;2 BYTE CRC
0006 =          FDRAL   EQU     6               ;LENGTH OF READ ADDRESS INFO

                ;  DISK HEADER SECTOR LENGTH VALUES
0000 =          FDSL128 EQU     0               ;SECTOR LENGTH 128
0001 =          FDSL256 EQU     1               ;SECTOR LENGTH 256
0002 =          FDSL512 EQU     2               ;SECTOR LENGTH 512
0003 =          FDSL1K  EQU     3               ;SECTOR LENGTH 1024

                ;  CONTROL REGISTER FLAGS
0001 =          CONIRQ  EQU     00000001B       ;ENABLE INT REQ
0002 =          CONDRQ  EQU     00000010B       ;ENABLE DRQ INT / DISABLE SYSTEM INT
0004 =          CONMFM  EQU     00000100B       ;ENABLE MFM
0008 =          CONMO   EQU     00001000B       ;MOTOR(S) ON
0010 =          CONDS0  EQU     00010000B       ;DRIVE 0
0020 =          CONDS1  EQU     00100000B       ;      1
0040 =          CONDS2  EQU     01000000B       ;      2
0080 =          CONDS3  EQU     10000000B       ;      3

                ;  MISCELLANEOUS VALUES
0028 =          NTRKS37 EQU     40              ;NUMBER OF TRACKS SINGLE DENSITY (48 TPI)
0050 =          NTRKD37 EQU     80              ;NUMBER OF TRACKS DOUBLE DENSITY (96 TPI)
000A =          NSPTS37 EQU     10              ;NUMBER OF SECTORS PER TRACK
                                                ;  SINGLE DENSITY
0010 =          NSPTD37 EQU     16              ;NUMBER OF SECTORS PER TRACK
                                                ;  DOUBLE DENSITY
0005 =          NSPTE37 EQU     5               ;NUMBER OF SECTORS PER TRACK
                                                ;  EXTENDED DOUBLE DENSITY
0003 =          ILFS37  EQU     3               ;INTERLEAVE FACTOR SINGLE DENSITY
0003 =          ILFD37  EQU     3               ;INTERLEAVE FACTOR DOUBLE DENSITY
0003 =          ILFE37  EQU     3               ;INTERLEAVE FACTOR EXTENDED DOUBLE DENSITY
003C =          NSBT37  EQU     60              ;NUMBER OF CP/M RECORDS TO BE LOADED AT BOOT
060F =          DELAY37 EQU     6*256+15        ;DESELECT AND MOTOR TURN OFF DELAY

0020 =          H37VEC  EQU     8*4             ;LEVEL 4 INTERRUPT
0023 =          DLYMO37 EQU     H37VEC+3        ;MOTOR TURN OFF DELAY COUNTER
```

```
0024 =      DLYH37  EQU     H37VEC+4        ;DESELECT DELAY COUNTER
0025 =      H37CTL  EQU     H37VEC+5        ;H37 CONTROL REGISTER IMAGE
0026 =      H37IRET EQU     H37VEC+6        ;WHERE TO GO AFTER INTERRUPT ADDRESS

                    PAGE
```

```
                        ;
                        ;  H67 EQUATES.
                        ;

0001 =                  H67BLKIO EQU    TRUE            ;TRUE  = USE Z80 BLOCK I/O
                                                        ;FALSE = USE HANDSHAKE I/O

0000 =                  SHUGART EQU     FALSE           ;TRUE  = HARD DISK IS SHUGART
                                                        ;FALSE = HARD DISK IS MEMOREX

                        ;  PORT DISPLACEMENTS.
0000 =                  HD$DAT  EQU     0               ;DATA PORT
0001 =                  HD$CON  EQU     1               ;CONTROL PORT
0001 =                  HD$STA  EQU     1               ;STATUS PORT
0002 =                  HD$SWI  EQU     2               ;SWITCH PORT

                        ;  CONTROL PORT FLAGS.
0080 =                  HDFACKH EQU     10000000B       ;ACKNOWLEDGE HOLD
0040 =                  HDFSEL  EQU     01000000B       ;SELECT
0020 =                  HDFEI   EQU     00100000B       ;ENABLE INTERRUPTS
0010 =                  HDFRES  EQU     00010000B       ;RESET
0002 =                  HDFDE   EQU     00000010B       ;DATA ENABLE

                        ;  BUS STATUS FLAGS.
0080 =                  HDBREQ  EQU     10000000B       ;REQUEST
0040 =                  HDBIO   EQU     01000000B       ;I/O (0=IN  1=OUT)
0020 =                  HDBMSG  EQU     00100000B       ;MSG
0010 =                  HDBCMD  EQU     00010000B       ;CMD/DATA (0=DATA  1=COMMNAD)
0008 =                  HDBBSY  EQU     00001000B       ;BUSY
0004 =                  HDBPE   EQU     00000100B       ;PARITY ERROR
0002 =                  HDBIRQ  EQU     00000010B       ;INTERRUPT REQUEST
0001 =                  HDBACK  EQU     00000001B       ;HOS IS HOLDING ACKNOWLEDGE LINE

                        ;  COMMANDS.
0000 =                  HDCTDR  EQU     000H            ;TEST DRIVE READY
0001 =                  HDCRCL  EQU     001H            ;RECALIBRATE
0002 =                  HDCRSY  EQU     002H            ;REQUEST SYNDROME
0003 =                  HDCRS   EQU     003H            ;REQUEST SENSE
0004 =                  HDCFD   EQU     004H            ;FORMAT DRIVE
0006 =                  HDCFT   EQU     006H            ;FORMAT TRACK
0007 =                  HDCFBS  EQU     007H            ;FORMAT BAD SECTOR
0008 =                  HDCRD   EQU     008H            ;READ
0009 =                  HDCWPS  EQU     009H            ;WRITE PROTECT SECTOR
000A =                  HDCWR   EQU     00AH            ;WRITE
000B =                  HDCSEK  EQU     00BH            ;SEEK
0020 =                  HDCCPY  EQU     020H            ;COPY
00C0 =                  HDCFDD  EQU     0C0H            ;FLOPPY DISK DESCRIPTION

                        ;  CLASS 0 COMMAND BLOCK STRUCTURE.
0000 =                  HDOOP   EQU     0               ;OPCODE
0001 =                  HDOLULA EQU     1               ;LOGICAL UNIT #/LOGICAL ADDR
00E0 =                  HDOLUN  EQU     11100000B       ;  BITS 7-5 = LOGICAL UNIT #
001F =                  HDOLA2  EQU     00011111B       ;  BITS 4-0 = LOGICAL ADDR (HIGH ORDER)
0002 =                  HDOLA1  EQU     2               ;LOGICAL ADDRESS (MIDDLE ORDER)
```

```
0003 =          HDOLAO  EQU     3               ;LOGICAL ADDRESS (LOW ORDER)
0004 =          HDONB   EQU     4               ;NUMBER OF BLOCKS / INTERLEAVE FACTOR
0005 =          HDOCON  EQU     5               ;CONTROL

                ;  CLASS 1 COMMAND BLOCK STRUCTURE.
0000 =          HD1OP   EQU     0               ;OPCODE
0001 =          HD1LUAS EQU     1               ;SOURCE LOGICAL UNIT #/ADDR
00E0 =          HD1LUNS EQU     11100000B       ;  BITS 7-5 = LOGICAL UNIT #
001F =          HD1LA2S EQU     00011111B       ;  BITS 4-0 = LOGICAL ADDR (HIGH ORDER)
0002 =          HD1LA1S EQU     2               ;SOURCE LOGICAL ADDRESS (MIDDLE ORDER)
0003 =          HD1LAOS EQU     3               ;SOURCE LOGICAL ADDRESS (LOW ORDER)
0004 =          HD1NB   EQU     4               ;NUMBER OF BLOCKS
0005 =          HD1LUAD EQU     5               ;DESTINATION LOGICAL UNIT #/ADDR
00E0 =          HD1LUND EQU     11100000B       ;  BITS 7-5 = LOGICAL UNIT #
001F =          HD1LA2D EQU     00011111B       ;  BITS 4-0 = LOGICAL ADDR (HIGH ORDER)
0006 =          HD1LA1D EQU     6               ;DESTINATION LOGICAL ADDR (MIDDLE ORDER)
0007 =          HD1LAOD EQU     7               ;DESTINATION LOGICAL ADDR (LOW ORDER)
0008 =          HD1SPAR EQU     8               ;SPARE
0009 =          HD1CON  EQU     9               ;CONTROL

                ;  CLASS 6 COMMAND BLOCK STRUCTURE.
0000 =          HD6OP   EQU     0               ;OPCODE
0001 =          HD6LUN  EQU     1               ;LOGICAL UNIT # (BITS 7-5)
0005 =          HD6TFC  EQU     5               ;TRACK FORMAT CODE

                ;  COMMAND BLOCK CONTROL BYTE FLAGS.
0080 =          HDFDR   EQU     10000000B       ;DISABLE RETRIES
0040 =          HDFDDEC EQU     01000000B       ;DISABLE DATE ERROR CORRECTION

                ;  TRACK FORMAT CODE FLAGS.
0002 =          HDFDEN  EQU     00000010B       ;DENSITY (0=SINGLE  1=DOUBLE)
0001 =          HDFSID  EQU     00000001B       ;SIDES (0=SINGLE  1=DOUBLE)

                ;  COMPLETION STATUS BYTE FLAGS.
00E0 =          HDFLUN  EQU     11100000B       ;LOGICAL UNIT # MASK
0002 =          HDFERR  EQU     00000010B       ;ERROR DURING COMMAND EXECUTION
0001 =          HDFPE   EQU     00000001B       ;PARITY ERROR

                ;  REQUEST SYNDROME BLOCK.
0000 =          HDSMBO  EQU     0               ;M.S. BIT OFFSET
0001 =          HDSLBS  EQU     1               ;L.S. BIT OFFSET / SYNDROME
00E0 =          HDFLBO  EQU     11100000B       ;  BITS 7-5 = L.S. BIT OFFSET
000F =          HDFSYN  EQU     00001111B       ;  BITS 3-0 = SYNDROME

                ;  REQUEST SENSE BLOCK.
0000 =          HDSSB   EQU     0               ;SENSE BYTE
0080 =          HDSBAV  EQU     10000000B       ;BLOCK ADDRESS VALID
0030 =          HDSET   EQU     00110000B       ;ERROR TYPE MASK
000F =          HDSEC   EQU     00001111B       ;ERROR CODE MASK
0001 =          HDSLULA EQU     1               ;LOGICAL UNIT #/LOGICAL ADDR
00E0 =          HDSLUN  EQU     11100000B       ;  BITS 7-5 = LOGICAL UNIT #
001F =          HDSLA2  EQU     00011111B       ;  BITS 4-0 = LOGICAL ADDR (HIGH ORDER)
0002 =          HDSLA1  EQU     2               ;LOGICAL ADDRESS (MIDDLE ORDER)
0003 =          HDSLAO  EQU     3               ;LOGICAL ADDRESS (LOW ORDER)

                ;  ERROR CODE TABLE.
```

```
                        ;    TYPE 0 (DRIVE) ERROR CODES
0000 =                  HDENS   EQU     000H            ;NO STATUS
0001 =                  HDENIS  EQU     001H            ;NO INDEX SIGNAL
0002 =                  HDENSC  EQU     002H            ;NO SEEK COMPLETE
0003 =                  HDEWF   EQU     003H            ;WRITE FAULT
0004 =                  HDEDNR  EQU     004H            ;DRIVE NOT READY
0005 =                  HDEDNS  EQU     005H            ;DRIVE NOT SELECTED
0006 =                  HDENTO  EQU     006H            ;NO TRACK 00
0007 =                  HDEMDS  EQU     007H            ;MULTI-DRIVE SELECTED
                        ;    TYPE 1 (CONTROLLER) ERROR CODES
0010 =                  HDEIR   EQU     010H            ;ID READ
0011 =                  HDEUD   EQU     011H            ;UNCORRECTABLE DATA
0012 =                  HDEIAM  EQU     012H            ;ID ADDRESS MARK NOT FOUND
0013 =                  HDEDAM  EQU     013H            ;DATA ADDRESS MARK NOT FOUND
0014 =                  HDERNF  EQU     014H            ;RECORD NOT FOUND
0015 =                  HDESE   EQU     015H            ;SEEK ERROR
0017 =                  HDEWP   EQU     017H            ;WRITE PROTECTED
0018 =                  HDECDF  EQU     018H            ;CORRECTABLE DATA FIELD ERROR
0019 =                  HDEBBF  EQU     019H            ;BAD BLOCK FOUND
001A =                  HDEFE   EQU     01AH            ;FORMAT ERROR
                        ;    TYPE 2 (COMMAND) ERROR CODES
0020 =                  HDEIC   EQU     020H            ;INVALID COMMAND
0021 =                  HDEIDA  EQU     021H            ;INVALID DISK ADDRESS
0022 =                  HDEIF   EQU     022H            ;ILLEGAL FUNCTION FOR THE DRIVE

                        ;  MISCELLANEOUS EQUATES

0122 =                  H67MIN  EQU     290             ;MINIMUM # OF SECTORS FOR A PARTITION
801A =                  H67MAX  EQU     32794           ;MAXIMUM # OF USEABLE SECTORS
                                                        ; FOR A PARTITION
                                IF      SHUGART
                        NSEC67  EQU     32768           ;TOTAL NUMBER OF SECTORS FOR SHUGART
                                ELSE
9880 =                  NSEC67  EQU     39040           ;TOTAL NUMBER OF SECTORS FOR MEMOREX
                                ENDIF
001A =                  NSPT67  EQU     26              ;NUMBER OF SECTORS PER TRACK
0001 =                  NSYS67H EQU     1               ;NUMBER OF BOOT TRACKS FOR HARD DISK

                                PAGE
```

```
                        ;
                        ;MISC EQUATES
                        ;

00C3 =          MI$JMP  EQU     0C3H            ;8080 JUMP INSTRUCTION

00F0 =          H8CTL   EQU     0F0H            ;H8 CONTROL PORT
00D0 =          H8TR    EQU     0D0H            ;H8 CLOCK TICK RESET

00F2 =          H88CTL  EQU     0F2H            ;H88 CONTROL PORT
0020 =          M1H     EQU     020H            ;KEEP RAM AT 0
0040 =          IOO     EQU     040H            ;H17 SIDE SELECT
0002 =          CLKE    EQU     002H            ;TURN ON 2MS CLOCK

0008 =          CLKVEC  EQU     0008H           ;CLOCK INTERRUPT VECTOR
000B =          TICCNT  EQU     000BH           ;TWO BYTE TICK COUNTER
000D =          CTLPRT  EQU     000DH           ;CURRENT CONTENTS OF '89 CONTROL LATCH
000E =          H8FLAG  EQU     000EH           ;CONTENTS = 0 FOR H/Z89, = H8TR FOR H8
000F =          DEVCTL  EQU     000FH           ;CURRENT CONTENTS OF H17 CONTROL LATCH
0018 =          SERVEC  EQU     0008H*3         ;SERIAL INTERRUPT VECTOR (LEVEL 3)

                        PAGE
```

```
                        ;
                        ;MISC CPM EQUATES
                        ;
0003 =                  NDISKS  EQU     H17ND+H37ND+H47ND+H67ND ;MAXIMUM NUMBER OF DISKS
                                                                ; IN THIS SYSTEM
                                IF      TRUE-1
                        %:      TRUE NE 1
                                ENDIF
                                IF      (8-NDISKS) SHR 15
                        %:      NDISK GT 8 -- DRIVE MAP ONLY HAS 8 ENTRY SLOTS
                                ENDIF

002C =                  NSECTS  EQU     44              ;NUM SECTS TO READ ON WM BOOT

                        ;  MAX HOST (PHYSICAL) SECTOR SIZE
                                IF      (H47T AND H47ED) OR (H37T AND H37ED)
                        HSTSIZ  EQU     1024            ;MAX HOST (PHYSICAL) SECTOR SIZE
                                ELSE
0100 =                  HSTSIZ  EQU     256             ;MAX HOST (PHYSICAL) SECTOR SIZE
                                ENDIF

00FF =                  BT$WM   EQU     0FFH            ;WARM BOOT FLAG
0000 =                  BT$CD   EQU     000H            ;COLD BOOT FLAG
                        ;
                        ;DEFAULT PORT ASSIGNMENTS
                        ;
00FA =                  H85CRT  EQU     372Q
00D0 =                  H84TTY  EQU     0D0H
00E8 =                  H84CRT  EQU     0E8H
00E0 =                  H84LPT  EQU     0E0H
00D8 =                  H84RDP  EQU     0D8H
                        ;
                        ;BAUD RATE DIVISORS FOR 8250'S
                        ;
0600 =                  B75     EQU     1536
0417 =                  B110    EQU     1047
0359 =                  B134    EQU     857
0180 =                  B300    EQU     384
00C0 =                  B600    EQU     192
0060 =                  B1200   EQU     96
0030 =                  B2400   EQU     48
0018 =                  B4800   EQU     24
000C =                  B9600   EQU     12
0006 =                  B19200  EQU     6
                        ;
                        ;ASCII VALUES
                        ;
0000 =                  NULL    EQU     00H
0003 =                  CTLC    EQU     03H
0007 =                  BELL    EQU     07H
000D =                  CR      EQU     0DH
000A =                  LF      EQU     0AH
00QD =                  PADCH   EQU     CR              ;CHAR THAT GETS NULL PADDING, MUST NOT BE NULL
```

```
                        ;DEFAULT I/O BYTE
                        ;       CON: = CRT:
                        ;       RDR: = UR1:
                        ;       PUN: = UP1:
                        ;       LST: = UL1:
                        ;

0000 =                  TTY     EQU     0

0001 =                  CRT     EQU     1
0001 =                  PTR     EQU     1
0001 =                  PTP     EQU     1

0002 =                  BAT     EQU     2
0002 =                  UR1     EQU     2
0002 =                  UP1     EQU     2
0002 =                  LPT     EQU     2

0003 =                  UC1     EQU     3
0003 =                  UR2     EQU     3
0003 =                  UP2     EQU     3
0003 =                  UL1     EQU     3

00A9 =                  DIOB    EQU     (CRT) OR (UR1 SHL 2) OR (UP1 SHL 4) OR (LPT SHL 6)

                                PAGE
```

```
                        ;
                        ;  DEVICE DRIVER ENTRY JUMP VECTOR OFFSETS.
                        ;

0000 =                  DDSEL   EQU     0               ;SELECT DISK
0003 =                  DDRD    EQU     3               ;READ
0006 =                  DDWR    EQU     6               ;WRITE
0009 =                  DDRES   EQU     9               ;RESET
000C =                  DDMNT   EQU     12              ;MOUNT

                        ;
                        ;  DISK PARAMETER ENTRY DESCRIPTION.
                        ;

0000 =                  DPEXLT  EQU     0               ;SECTOR TRANSLATE TABLE ADDRESS
000A =                  DPEDPB  EQU     10              ;DISK PARAMETER BLOCK ADDRESS
0010 =                  DPEHTH  EQU     16              ;HEATH EXTENSIONS
0018 =                  DPEL    EQU     24              ;LENGTH OF DISK PARAMETER ENTRY

                        ;  HEATH EXTENSIONS.
0010 =                  DPEFLAG EQU     DPEHTH+0        ;FLAGS
00E0 =                  DPETYPE EQU     11100000B       ;BIT 7-5 = DEVICE TYPE
0000 =                  DPENE   EQU     00000000B       ;NON-EXISTENT
0040 =                  DPEH17  EQU     01000000B       ;H17
0060 =                  DPEH37  EQU     01100000B       ;H37
0080 =                  DPEH47  EQU     10000000B       ;H47
00C0 =                  DPEH67H EQU     11000000B       ;H67 HARD DISK
00C8 =                  DPEH67F EQU     11001000B       ;H67 FLOPPY DISK
00E8 =                  DPETYPF EQU     DPETYPE OR DPEH67F      ;H67 TYPE MASK
0010 =                  DPEP7C  EQU     00010000B       ;BIT 4 -- 0=BASE PORT 78H  1=PORT 7CH
0010 =                  DPE48RO EQU     00010000B       ;BIT 4 -- FOR H37
                                                        ;  48 TPI MEDIA IN 96 TPI DRIVE (R/O)
0008 =                  DPE96T  EQU     00001000B       ;BIT 3 -- 0=48 TPI DRIVE  1=96 TPI DRIVE
0004 =                  DPEASGN EQU     00000100B       ;BIT 2 -- FOR H67 HARD DISK
                                                        ;         0=UNASSIGNED A PARTITION
                                                        ;         1=ASSIGNED A PARTITION
0004 =                  DPEED   EQU     00000100B       ;BIT 2 -- 1=EXTENDED DOUBLE DENSITY
0002 =                  DPEDD   EQU     00000010B       ;BIT 1 -- 0=SINGLE DENSITY  1=DOUBLE
0001 =                  DPE2S   EQU     00000001B       ;BIT 0 -- 0=SINGLE SIDED  1=DOUBLE

0011 =                  DPEUNIT EQU     DPEHTH+1        ;UNIT SELECT VALUE
0012 =                  DPERPS  EQU     DPEHTH+2        ;CP/M RECORDS PER PHYSICAL SECTOR
0013 =                  DPERPAB EQU     DPEHTH+3        ;CP/M RECORDS PER ALLOCATION BLOCK
0014 =                  DPETRK  EQU     DPEHTH+4        ;TRACK COUNTER
0080 =                  DPEUNK  EQU     10000000B       ;TRACK POSITION UNKNOWN
0015 =                  DPESEK  EQU     DPEHTH+5        ;MOTOR SPEED AND SEEK SPEED
                                                        ;BIT 6-0 = SEEK SPEED VALUE
0080 =                  DPEMO   EQU     10000000B       ;BIT 7 = MOTOR UP TO SPEED FLAG
                                                        ;  0=1 SEC  1=250 MSEC
0016 =                  DPEUPB  EQU     DPEHTH+6        ;H67 PARTITION UPPER BOUND + 1
0016 =                  DPEFLG2 EQU     DPEHTH+6        ;2ND FLAG BYTE
0002 =                  DPEIMG  EQU     00000010B       ;BIT 1  IMAGINARY DRIVE
0001 =                  DPE96TM EQU     00000001B       ;BIT 0  0=48 TPI MEDIA  1=96 TPI MEDIA
0017 =                  DPELUN  EQU     DPEHTH+7        ;LAST LOGICAL UNIT MOUNTED
```

```
0008 =          DPEHL   EQU     8               ;LENGTH OF HEATH EXTENSION

                ;
                ;  DISK PARAMETER BLOCK.
                ;

0000 =          DPBSPT  EQU     0               ;SECTORS PER TRACK
0002 =          DPBBSH  EQU     DPBSPT+2        ;BLOCK SHIFT FACTOR
0003 =          DPBBLM  EQU     DPBSPT+3        ;BLOCK MASK
0004 =          DPBEXM  EQU     DPBSPT+4        ;EXTENT MASK
0005 =          DPBDSM  EQU     DPBSPT+5        ;TOTAL # OF BLOCKS - 1
0007 =          DPBDRM  EQU     DPBSPT+7        ;# OF DIRECTORY ENTRIES - 1
0009 =          DPBAL0  EQU     DPBSPT+9        ;INITIAL AL0 VALUE
000A =          DPBAL1  EQU     DPBSPT+10       ;INITIAL AL1 VALUE
000B =          DPBCKS  EQU     DPBSPT+11       ;SIZE OF DIRECTORY CHECK VECTOR
000D =          DPBOFF  EQU     DPBSPT+13       ;NUMBER OF SYSTEM TRACKS
000F =          DPBL    EQU     15              ;LENGTH OF DISK PARAMETER BLOCK

                ;********************
                ;
                ;  DISK LABEL DEFINITIONS
                ;
                ;  DISK LABELS ARE USED ON ALL SYSGEN'ED DISKS.
                ;  ALSO THE H37 AND H67 HARD DATA DISKS USE THE LABEL.
                ;  TO MAINTAIN COMPATIBILITY WITH OLDER RELEASES OF CP/M
                ;  THE H17, H47, AND H67 FLOPPY DATA DISKS DO NOT USE LABELS.
                ;
                ;  THE LABEL RESIDES ON THE 1ST SECTOR OF TRACK 0.
                ;
                ;  AT THE END OF THE LABEL IS A CHECKSUM.  THE CHECKSUM IS CALCULATED
                ;  BY ADDING UP THE VALUES IN THE LABEL PRIOR TO THE CHECKSUM SLOT
                ;  A BYTE AT A TIME, THEN TAKING THE ONE'S COMPLEMENT OF THE SUM.
                ;
                ;********************

0000 =          LABVER  EQU     0               ;CURRENT FORM # FOR LABEL

0000 =          LABBUF  EQU     0               ;SLOT FOR JUMP INSTRUCTION AROUND LABEL
0003 =          BDTYPE  EQU     LABBUF+3        ;SLOT FOR DRIVE TYPE

0004 =          LABEL   EQU     LABBUF+4
0004 =          LABTYP  EQU     LABEL+0         ;SLOT FOR LABEL TYPE
0005 =          LABHTH  EQU     LABTYP+1        ;SLOT FOR HEATH EXTENSIONS TO DPE
000D =          LABDPB  EQU     LABHTH+DPEHL    ;SLOT FOR DISK PARAMETER BLOCK
001C =          LABCS   EQU     LABDPB+DPBL     ;CHECKSUM

0019 =          LABLEN  EQU     LABCS-LABEL+1   ;LABEL LENGTH

                        PAGE
```

```
                        ;  SOFTWARE BOOT CODE DEFINITIONS

0000 =                  SBC$SBC EQU     0               ;SECTOR # OF SBC

0000 =                  SBC$JMP EQU     0000H           ;JUMP TO SOFTWARE BOOT CODE
0003 =                  SBC$VER EQU     0003H           ;SOFTWARE BOOT CODE VERSION NUMBER
0004 =                  SBC$REV EQU     0004H           ;SOFTWARE BOOT CODE REVISION NUMBER
0005 =                  SBC$DBS EQU     0005H           ;DEFAULT BOOT STRING
0018 =                  SBC$BSA EQU     0018H           ;SECTOR ADDR OF BAD SECTOR TABLE A
001B =                  SBC$BSB EQU     001BH           ;SECTOR ADDR OF BAD SECTOR TABLE B
001E =                  SBC$SBA EQU     001EH           ;SECTOR ADDR OF SUPER BLOCK A
0021 =                  SBC$SBB EQU     0021H           ;SECTOR ADDR OF SUPER BLOCK B
0024 =                  SBC$SSZ EQU     0024H           ;SECTOR SIZE
0026 =                  SBC$SPT EQU     0026H           ;SECTORS PER TRACK
0028 =                  SBC$TPC EQU     0028H           ;TRACKS PER CYLINDER
002A =                  SBC$CPV EQU     002AH           ;CYLINDERS PER VOLUME
002C =                  SBC$SPS EQU     002CH           ;SECTORS PER SLAB
002E =                  SBC$VSZ EQU     002EH           ;VOLUME SIZE (SECTORS PER VOLUME)
0031 =                  SBC$NSL EQU     0031H           ;NUMBER OF SLABS - 1
0032 =                  SBC$CSA EQU     0032H           ;CHECKSUM: SUPER BLOCK A
0034 =                  SBC$CSB EQU     0034H           ;CHECKSUM: SUPER BLOCK B
0036 =                  SBC$CBA EQU     0036H           ;CHECKSUM: BAD SECTOR TABLE A
0038 =                  SBC$CBB EQU     0038H           ;CHECKSUM: BAD SECTOR TABLE B

0080 =                  SBC$LEN EQU     0080H           ;SBC LENGTH

                        ;  SUPER BLOCK DEFINITIONS

0000 =                  SPB$OSD EQU     0               ;OPERATING SYSTEM DEFINITIONS
0001 =                  SPB$PAT EQU     1               ;PARTITION TABLE

                        ;    OPERATING SYSTEM NAME TABLE DEFINITIONS

0010 =                  SPB$OSL EQU     16              ;LENGTH OF EACH ENTRY
0010 =                  SPB$OSN EQU     16              ;NUMBER OF ENTRIES

                        ;    PARTITION TABLE ENTRIES

0000 =                  SPB$OSI EQU     0               ;OPERATING SYSTEM ID
001F =                  SPB$OSM EQU     00011111B       ;OPERATING SYSTEM ID MASK
001E =                  SPB$UAR EQU     00011110B       ;UNALLOCATED REGION
001F =                  SPB$EOL EQU     SPB$OSM         ;END OF LIST

0001 =                  SPB$FSN EQU     1               ;FIRST SECTOR # OF PARTITION
                                                        ; (LOW,MID,HIGH)

0004 =                  SPB$PEL EQU     4               ;LENGTH OF ENTRY

                        ;  BAD SECTOR TABLE

0003 =                  SBC$BEL EQU     3               ;LENGTH OF ENTRY

                                PAGE
```

```
                        ;
                        ; ENTRY POINT TABLE
                        ;
0000 C3340C             JMP     CBOOT           ;FROM COLD START LOADER
0003 C39600     WBOOTE: JMP     WBOOT           ;TO INITIATE A WARM BOOT
0006 C3CE08             JMP     CONST           ;CHECK CONSOLE STATUS
0009 C3F308             JMP     CONIN           ;READ CONSOLE CHAR
000C C30109             JMP     CONOUT          ;WRITE CONSOLE CHAR
000F C31F09             JMP     LIST            ;WRITE LIST DEVICE CHAR
0012 C32F09             JMP     PUNCH           ;WRITE PUNCH DEVICE CHAR
0015 C34009             JMP     READER          ;READ CHAR FROM READER
0018 C37701             JMP     HOME            ;SET DISK TO TRACK ZERO
001B C38A01             JMP     SETDSK          ;SELECT DISK DRIVE
001E C38401             JMP     SETTRK          ;SEEK TO TRACK
0021 C31202             JMP     SETSEC          ;SET SECTOR NUMBER
0024 C31802             JMP     SETDMA          ;SET STARTING ADDRESS FOR DISK I/O
0027 C32B02             JMP     READ            ;READ SELECTED SECTOR
002A C33F02             JMP     WRITE           ;WRITE SELECTED SECTOR
002D C30F09             JMP     LISTST          ;CHECK LIST DEVICE STATUS
0030 C31E02             JMP     SECTRAN         ;SECTOR TRANSLATE ROUTINE

                        PAGE
```

```
                        ;*******************
                        ;
                        ;  BIOS HEADER SECTION.
                        ;
                        ;  * * *   N O T E   * * *
                        ;
                        ;     IF THIS SECTION IS ALTERED, THEN THE FILE 'BIOSDEF.LIB'
                        ;     MUST ALSO BE UPDATED.
                        ;
                        ;*******************

0033 03                 BIOSVER DB      VERS            ;BIOS VERSION
0034 A9                 DEFIOB  DB      DIOB            ;DEFAULT IOBYTE
0035 10                 PRTRDY  DB      010H            ;PRINTER (LPT) READY MASK
                                                        ; (CTS = 010H WHICH IS DEFAULT
                                                        ;  DSR = 020H)

0036 80                 MODE    DB      PARTITN*MODEB3+MODEB7
0001 =                  MODEB0  EQU     00000001B       ;BIT 0  1=CRT ON H8-5 CARD
0002 =                  MODEB1  EQU     00000010B       ;BIT 1  1=EXTENDED DISK ERROR MSG'S
0004 =                  MODEB2  EQU     00000100B       ;BIT 2  0 = LPT READY IS LOW
                                                        ;       1 = LPT READY IS HIGH
0008 =                  MODEB3  EQU     00001000B       ;BIT 3  1=PARTITIONING USED
0040 =                  MODEB6  EQU     01000000B       ;BIT 6  1=RUN 'AUTO' ON WARM BOOT
0080 =                  MODEB7  EQU     10000000B       ;BIT 7  1=RUN 'AUTO' ON COLD BOOT

                        ; SERIAL DEVICE STRUCTURES
                        ;       DB      PORT$NUMBER
                        ;       DW      CONTROL$WORD
                        ;       WHERE THE CONTROL$WORD CONTAINS
                        ;               B15     MAP LOWER TO UPPER CASE
                        ;               B14-B12 NUMBER OF NULLS AFTER A CR
                        ;               B11-B00 BAUD RATE DIVISOR

0037 E8                 H84PT1: DB      H84CRT
0038 0C80               CRTBAUD DW      B9600+8000H
003A D0                 H84PT2: DB      H84TTY
003B 8001               TTYBAUD:DW      B300
003D E0                 H84PT3: DB      H84LPT
003E 1800               LPTBAUD:DW      B4800
0040 D8                 H84PT4: DB      H84RDP
0041 8001               RDPBAUD:DW      B300
0043 00                         DB      0               ;RESERVED FOR 5TH SERIAL DEVICE STRUCTURE
0044 0000                       DW      0
                        ;
0046 10                 BSIZE   DB      (BIOSEND-BIOS+255)/256  ;BIOS SIZE IN PAGES
0047 050F               BEND    DW      BIOSEND         ;ENDING ADDRESS OF BIOS + 1
0049 0000               SECNT17 DW      0               ;H17 SOFT ERROR COUNT (SINCE COLD BOOT)
004B 0000               SECNT37 DW      0               ;H37 SOFT ERROR COUNT (SINCE COLD BOOT)
004D 03                 BNDISKS DB      NDISKS          ;NUMBER OF DISKS POSSIBLE IN THIS BIOS

004E                    DPBASE  DS      0               ;START OF DISK PARAMETER ENTRY TABLES

                                PAGE
```

```
                        IF      H17T
004E 1D0C0000  DPE0:    DW      XLT17,0000H     ;TRANSLATE TABLE
0052 00000000           DW      0000H,0000H     ;SCRATCH
0056 340D0E0C           DW      DIRBUF,DPB17S   ;DIR BUFF,PARM BLK
005A C00DB40D           DW      CSV0,ALV0       ;CHECK,ALLOC VEC
005E 40                 DB      DPEH17          ;DISK TYPE
005F 02                 DB      U0              ;SELECT CODE
0060 02                 DB      2               ;NUMBER OF 128 BYTE RECORDS/PHYSICAL SECTOR
0061 08                 DB      8               ;NUMBER OF RECORDS/ALLOCATION BLOCK
0062 FF                 DB      0FFH            ;TRACK LOCATION
0063 0F                 DB      STEPR           ;STEP RATE
0064 00                 DB      0               ;FLAG BYTE 2
0065 00                 DB      0               ;'REAL' - LAST DISK MOUNTED
                                                ;'IMAGINARY' - CORRESPONDING REAL
                                                ;   DRIVE'S LOGICAL UNIT #
0066 1D0C0000  DPE1:    DW      XLT17,0000H
006A 00000000           DW      0000H,0000H
006E 340D0E0C           DW      DIRBUF,DPB17S
0072 DC0DD00D           DW      CSV1,ALV1
0076 40                 DB      DPEH17
0077 04                 DB      U1
0078 02                 DB      2
0079 08                 DB      8
007A FF                 DB      0FFH
007B 0F                 DB      STEPR
007C 00                 DB      0
007D 00                 DB      0
007E 1D0C0000  DPE2:    DW      XLT17,0000H
0082 00000000           DW      0000H,0000H
0086 340D0E0C           DW      DIRBUF,DPB17S
008A F80DEC0D           DW      CSV2,ALV2
008E 40                 DB      DPEH17
008F 08                 DB      U2
0090 02                 DB      2
0091 08                 DB      8
0092 FF                 DB      0FFH
0093 0F                 DB      STEPR
0094 00                 DB      0
0095 00                 DB      0
                        ENDIF

                        IF      H37T
                DPE37$0 DW      0000H,0000H     ;TRANSLATE TABLE
                        DW      0000H,0000H     ;SCRATCH
                        DW      DIRBUF,DPB37$0  ;DIR BUFF, PARM BLK
                        DW      CSV37$0,ALV37$0
                        DB      DPEH37+DPEDD    ;ASSUME DOUBLE DENSITY AT COLD BOOT
                        DB      CONDS0
                        DB      2
                        DB      8
                        DB      DPEUNK
                        DB      FDFS30
                        DB      0
                        DB      0
                DPE37$1 DW      0000H,0000H
```

```
                DW      0000H,0000H
                DW      DIRBUF,DPB37$1
                DW      CSV37$1,ALV37$1
                DB      DPEH37+DPEDD
                DB      CONDS1
                DB      2
                DB      8
                DB      DPEUNK
                DB      FDFS30
                DB      0
                DB      0
DPE37$2         DW      0000H,0000H
                DW      0000H,0000H
                DW      DIRBUF,DPB37$2
                DW      CSV37$2,ALV37$2
                DB      DPEH37+DPEDD
                DB      CONDS2
                DB      2
                DB      8
                DB      DPEUNK
                DB      FDFS30
                DB      0
                DB      0
                ENDIF

                IF      H47T
DPE47$0         DW      XLTOS,0000H     ;TRANSLATE TABLE
                DW      0000H,0000H     ;SCRATCH
                DW      DIRBUF,DPBOSS   ;DIR BUFF, PARM BLK
                DW      CSV47$0,ALV47$0 ;CHECK, ALLOC VEC
                DB      DPEH47
                DB      000H
                DB      1
                DB      8
                DB      0,0,0,0
DPE47$1         DW      XLTOS,0000H
                DW      0000H,0000H
                DW      DIRBUF,DPBOSS
                DW      CSV47$1,ALV47$1
                DB      DPEH47
                DB      020H
                DB      1
                DB      8
                DB      0,0,0,0
                ENDIF

                IF      H67T
DPE67$0         DW      0000H,0000H
                DW      0000H,0000H
                DW      DIRBUF,DPB67$0
                DW      0000H,ALV67$0
                DB      DPEH67H         ;H67 HARD DISK
                DB      0               ;SELECT
                DB      2               ;LOGICAL IN EACH PHYSICAL
                DB      32              ;ALLOCATED AT ONCE
                DW      0               ;TRACK 0 OFFSET
```

```
                DW      0               ;END OF PARTITION + 1
                IF      H67PART2
DPE67$2         DW      0000H,0000H
                DW      0000H,0000H
                DW      DIRBUF,DPB67$2
                DW      0000H,ALV67$2
                DB      DPEH67H         ;H67 HARD DISK (2ND PARTITION)
                DB      0               ;SELECT
                DB      2               ;LOGICAL IN EACH PHYSICAL
                DB      32              ;ALLOCATED AT ONCE
                DW      0               ;TRACK 0 OFFSET
                DW      0               ;END OF PARTITION + 1
                ENDIF
DPE67$1         DW      XLTOS,0000H     ;FLOPPY IN H67
                DW      0000H,0000H
                DW      DIRBUF,DPBOSS
                DW      CSV67$1,ALV67$1
                DB      DPEH67F         ;H67 FLOPPY
                DB      020H            ;SELECT
                DB      1               ;LOGICAL IN EACH PHYSICAL
                DB      8               ;ALLOCATED AT ONCE
                DB      0,0,0,0
                ENDIF

                PAGE
```

```
                        ;
                        ;       WARM BOOT -- READ IN BDOS AND CCP
                        ;                    INITIALIZE
                        ;                    JUMP TO CCP
                        ;

0096 31050F     WBOOT:  LXI     SP,STACK        ;SET STACK POINTER
0099 FB                 EI                      ;INSURE INTERRUPTS ARE ENABLED

009A AF                 XRA     A
009B 4F                 MOV     C,A             ;BOOT FROM DRIVE 0
009C 5F                 MOV     E,A             ;ACT LIKE THIS IS THE FIRST LOGIN
009D CD8A01             CALL    SETDSK
00A0 7C                 MOV     A,H
00A1 B5                 ORA     L
00A2 CA5F01             JZ      WBTE            ; BR IF SELECT ERROR
00A5 E5                 PUSH    H               ;GET POINTER TO XLATE AND SPT VALUE FOR
                                                ; TRACKS 1 TO N
00A6 CDB308             CALL    HLIHL           ;GET THE POINTER TO THE TRANSLATE TABLE
00A9 22120E             SHLD    XLTW1           ; AND SAVE IT
00AC 220F0E             SHLD    XLTW
00AF E1                 POP     H
00B0 E5                 PUSH    H
00B1 110A00             LXI     D,DPEDPB        ;GET THE POINTER TO THE DISK PARAM BLOCK
00B4 19                 DAD     D
00B5 CDB308             CALL    HLIHL
00B8 7E                 MOV     A,M             ;THE FIRST ENTRY OF WHICH IS THE NUMBER
00B9 32110E             STA     SPT1            ; OF SECTORS PER TRACK
00BC 320E0E             STA     SPT
00BF E1                 POP     H

                        IF      H47T OR H67T
                        PUSH    H
                        LXI     D,DPEHTH
                        DAD     D               ;GET POINTER TO HEATH DISK TABLE
                        MOV     A,M             ;GET TYPE BYTE
                        MOV     C,A
                        ANI     DPETYPE
                        CPI     DPEH47          ;CHECK FOR TYPE H47
                        JZ      WBTOX           ;IS H47
                        MOV     A,C
                        ANI     DPETYPF
                        CPI     DPEH67F         ;CHECK FOR TYPE H67 FLOPPY
                        JNZ     WBTO            ;IF NOT H47 OR H67, THEN USE THESE VALUES
                WBTOX:  MVI     A,26            ;  ELSE USE SINGLE DENSITY VALUES
                        LXI     H,XLTOS         ;    FOR TRACK 0
                        STA     SPT
                        SHLD    XLTW
                WBTO:   POP     H
                        ENDIF

00C0 111000             LXI     D,DPEHTH
00C3 19                 DAD     D               ;GET POINTER TO HEATH DISK TABLE
00C4 1180E9             LXI     D,CCP-128       ;ASSUME STARTING ADDRESS OF BOOT TRACK
00C7 01002C             LXI     B,NSECTS*256    ;B = # OF SECTORS, C = STARTING SECTOR
```

```
                        IF      H37T OR H67T
                        MOV     A,M             ;GET TYPE BYTE
                        ANI     DPETYPE
                        CPI     DPBH37          ;CHECK FOR H37
                        JZ      WBTOY           ;BR IF H37
                        CPI     DPEH67H         ;CHECK FOR H67
                        JNZ     WBTOZ           ;BR IF NOT H67
                WBTOY:  LXI     D,CCP-256       ;RESET STARTING ADDRESS OF BOOT TRACK
                        ENDIF

00CA 210000     WBTOZ:  LXI     H,0             ;STARTING TRACK = 0

00CD 221E0E     WBT1:   SHLD    SEKTRK
00D0 EB                 XCHG
00D1 220C0E             SHLD    DMAB            ;STARTING DMA ADR FOR THIS TRACK

00D4 C5         WBT2:   PUSH    B
00D5 0600               MVI     B,0             ;TRANSLATE SECTOR IN BC
00D7 2A0F0E             LHLD    XLTW
00DA EB                 XCHG
00DB CD1E02             CALL    SECTRAN
00DE 4D                 MOV     C,L             ;TRANSLATED SECTOR IN HL
00DF 7D                 MOV     A,L
00E0 F5                 PUSH    PSW
00E1 CD1202             CALL    SETSEC          ;SET THIS AS THE SECTOR TO READ
00E4 F1                 POP     PSW
00E5 3D                 DCR     A               ;CHANGE SECTOR NUMBER TO 0 THRU SPT-1
00E6 CD6B01             CALL    CDA             ;CORRECT DMA ADDRESS FOR THIS SECTOR
00E9 222E0E             SHLD    DMAADR
00EC 7C                 MOV     A,H
00ED FEEA               CPI     CCP/256         ;Q. IS THIS SECTOR PART OF BOOT CODE
00EF DA0401             JC      WBT3            ;YES, SO DON'T REALLY READ IT
00F2 FE00               CPI     BIOS/256        ;IS THIS SECTOR PART OF THE BIOS
00F4 D20401             JNC     WBT3            ;YES, SO DON'T REALLY READ IT

00F7 CD2B02             CALL    READ            ;READ THIS SECTOR

00FA B7                 ORA     A               ;CHECK FOR ERRORS
00FB C25F01             JNZ     WBTE

00FE C1                 POP     B
00FF 05                 DCR     B               ;COUNT THIS SECTOR AS READ
0100 CA2801             JZ      WBT4            ;IF THAT IS THE LAST ONE, SET POINTERS & LEAVE

0103 C5                 PUSH    B
0104 C1         WBT3:   POP     B
0105 0C                 INR     C               ;NEXT SECTOR
0106 3A0E0E             LDA     SPT             ;NUMBER OF SECTORS PER TRACK
0109 B9                 CMP     C               ;HAVE WE OVERFLOWED TO NEXT TRACK?
010A C2D400             JNZ     WBT2            ;NO
010D 0E00               MVI     C,0
010F CD6B01             CALL    CDA             ;UPDATE TRACK STARTING ADDRESS

0112 E5                 PUSH    H
0113 3A110E             LDA     SPT1            ;BEYOND TRACK 0,
```

```
0116 320E0E            STA     SPT             ; SO UPDATE SPT AND XLTW TO BE THE VALUES
0119 2A120E            LHLD    XLTW1           ; FOR TRACKS 1 AND BEYOND
011C 220F0E            SHLD    XLTW
011F E1                POP     H

0120 EB                XCHG
0121 2A1E0E            LHLD    SEKTRK
0124 23                INX     H
0125 C3CD00            JMP     WBT1

0128 3EFF      WBT4:   MVI     A,BT$WM         ;FLAG THIS AS A WARM BOOT

012A F5        GOW:    PUSH    PSW             ;SAVE THE BOOT TYPE

012B 3EC3              MVI     A,MI$JMP        ;INITIALIZE BIOS AND BDOS VECTORS
012D 210300            LXI     H,WBOOTE
0130 320000            STA     BOOT
0133 220100            SHLD    BOOT+1
0136 2106F2            LXI     H,BDOS+6
0139 320500            STA     BOOT+5
013C 220600            SHLD    BOOT+6

013F 018000            LXI     B,BUFF          ;SET DEFAULT DMA ADDRESS
0142 CD1802            CALL    SETDMA

0145 CD0E04            CALL    FLUSH1          ;RE-INIT HOST DEBLOCKING

0148 F1                POP     PSW             ;GET THE BOOT TYPE
0149 0F                RRC                     ; CARRY SET IF WARM BOOT
014A 3A3600            LDA     MODE
014D D25101            JNC     GOW1            ;IF WARM BOOT
0150 17                RAL                     ; THEN SHIFT LEFT TWICE
0151 17        GOW1:   RAL                     ;CARRY SET IF TO RUN AUTO
0152 3A0400            LDA     LOGDSK          ;GET DISK NUMBER TO
0155 321D0E            STA     SEKDSK          ; SAVE AS DESIRED DISK
0158 4F                MOV     C,A             ; PASS TO CCP IN C
0159 DA00EA            JC      CCP             ;EXECUTE AUTO
015C C303EA            JMP     CCPCLR          ;EXECUTE CCP

015F 21C20B    WBTE:   LXI     H,BTMSG         ;PRINT BOOT ERROR MESSAGE
0162 CD9D0B            CALL    PMSG
0165 CDF308            CALL    CONIN           ;WAIT FOR KEYBOARD
0168 C39600            JMP     WBOOT           ;TRY AGAIN

                       PAGE
```

```
                        ;CDA - CORRECT DMA ADDRESS
                        ;       ENTRY   A       = RECORDS TO ADJUST
                        ;               DMAB    = STARTING ADDRESS OF TRACK
                        ;       EXIT    HL      = CORRECTED ADDRESS
                        ;

016B 2A0C0E     CDA:    LHLD    DMAB            ;GET STARTING ADDRESS OF TRACK
016E B7                 ORA     A               ;CLEAR CARRY
016F 1F                 RAR                     ;DIVIDE BY 2
0170 57                 MOV     D,A             ;  D = A/2
0171 3E00               MVI     A,0
0173 1F                 RAR
0174 5F                 MOV     E,A             ;  E = 00H OR 80H
0175 19                 DAD     D
0176 C9                 RET

                        PAGE
```

```
                        ;
                        ;       HOME - SEEK HEAD TO TRACK 0 (DEFERRED)
                        ;

0177 3A320C     HOME:   LDA     HSTWRT          ;CHECK FOR PENDING WRITE
017A B7                 ORA     A
017B C28101             JNZ     HOMED
017E 32310C             STA     HSTACT          ;CLEAR HOST ACTIVE FLAG
0181 010000     HOMED:  LXI     B,0             ;SET TO TRACK 0

                        ;
                        ;       SETTRK - SET DISK TRACK NUMBER
                        ;               ENTRY:  BC      DESIRED TRACK ( 0 TO N-1 )
                        ;

0184 60         SETTRK: MOV     H,B
0185 69                 MOV     L,C
0186 221E0E             SHLD    SEKTRK
0189 C9                 RET

                        PAGE
```

```
                        ;
                        ;       SETDSK - SELECT A DISK DRIVE
                        ;               ENTRY:  C       DESIRED DISK
                        ;                       E       LSB = 0 IF FIRST LOGIN
                        ;
018A 7B         SETDSK: MOV     A,E
018B 32EC01             STA     SETDSKB         ;SAVE 1ST FLAG
018E 79                 MOV     A,C             ;GET THE DRIVE # FROM C
018F 32EB01             STA     SETDSKA         ;SAVE LOGICAL UNIT NAME
0192 FE03               CPI     NDISKS          ;CHECK IF LEGAL
0194 D2DB01             JNC     SETDE           ; BR IF NOT

0197 214000             LXI     H,BDMAP         ;GET MAPPED DRIVE #
019A CD9508             CALL    DADA            ;(HL)=POINTER TO LOGICAL/MAPPED DRIVE #
019D 7E                 MOV     A,M             ;GET LOGICAL/MAPPED DRIVE #
019E 321D0E             STA     SEKDSK          ;SAVE IT
01A1 CD9A08             CALL    GETDPE          ;GET ADDR OF DPE
01A4 22ED01             SHLD    SETDSKC         ;SAVE ADDR OF DPE
01A7 111000             LXI     D,DPEHTH        ;GET ADDR OF DPE'S HEATH EXTENSIONS
01AA 19                 DAD     D
01AB 22080E             SHLD    DPBX            ;SAVE IT

                        IF      PARTITN
                        MOV     A,M             ;GET HEATH EXTENSION FLAG BYTE
                        ANI     DPETYPF
                        CPI     DPEH67H
                        JNZ     SETDSK1         ;BR IF NOT H67 HARD DISK
                        MOV     A,M
                        ANI     DPEASGN         ;CHECK IF PARTITION IS ASSIGNED
                        JZ      SETDE           ;BR IF NOT
                        ENDIF

                SETDSK1:
01AE 3AEC01             LDA     SETDSKB         ;RESTORE FIRST LOGIN FLAG
01B1 1F                 RAR                     ;IF FIRST LOGIN
01B2 DAD701             JC      SETDSK2         ; BR IF NOT

01B5 CD0004             CALL    FLUSH           ;FLUSH HOST BUFFER (I MAY NEED IT
                                                ; AND ITS ASSOCIATED VARIABLES)

01B8 213600             LXI     H,MODE          ;SAVE MODE
01BB 7E                 MOV     A,M
01BC 32EF01             STA     SETDSKD
01BF E6FD               ANI     OFFH-MODEB1     ;TURN OFF EXTENDED ERROR MSG'S
01C1 77                 MOV     M,A

01C2 3A1D0E             LDA     SEKDSK          ;SET HOST DPE'S HEATH EXTENSION
01C5 CD4D04             CALL    SHD             ; ADDR AND SWAP DISK IF NEEDED

01C8 110000             LXI     D,DDSEL         ;CALL DEVICE DRIVER FOR SELECT
01CB CDF001             CALL    DSKDIS

01CE 3AEF01             LDA     SETDSKD         ;RESTORE BIOS MODE BYTE
01D1 323600             STA     MODE
```

```
01D4 DADB01            JC      SETDE          ; BR IF ERROR RETURNED BY DEVICE DRIVER

               SETDSK2:
01D7 2AED01            LHLD    SETDSKC        ;RET WITH ADDR OF DPE
01DA C9                RET

01DB 3AEB01    SETDE:  LDA     SETDSKA        ;GET DRIVE # FROM ENTRY
01DE 210400            LXI     H,LOGDSK       ;CHECK TO SEE IF ERROR IS
01E1 BE                CMP     M              ; ON DEFAULT DISK
01E2 C2E701            JNZ     SETDE1         ;  BR IF NOT
01E5 3600              MVI     M,0            ;SET DEFAULT AS DRIVE A
01E7 210000    SETDE1: LXI     H,0000H        ;SET ERROR RETURN VALUE
01EA C9                RET

01EB           SETDSKA DS      1              ;LOGICAL UNIT # FOR SELECTION
01EC           SETDSKB DS      1              ;1ST TIME FLAG
01ED           SETDSKC DS      2              ;ADDR RETURNED BY GETDPE
01EF           SETDSKD DS      1              ;SAVED BIOS MODE BYTE

                       PAGE
```

```
01F0 7E          DSKDIS: MOV     A,M             ;GET DISK DRIVE TYPE
01F1 07                  RLC
01F2 07                  RLC
01F3 07                  RLC
01F4 E607                ANI     DPETYPE/32
                         IF      DPETYPE-11100000B
                 $:      DPETYPE NE 11100000B
                         ENDIF
01F6 87                  ADD     A               ;*2
01F7 210202              LXI     H,DTT
01FA CD9508              CALL    DADA
01FD CDB308              CALL    HLIHL
0200 19                  DAD     D               ;ADD JUMP VECTOR OFFSET
0201 E9                  PCHL

                 DTT:
0202 FA07                DW      NULDVD          ;000 - NON-EXISTANT
0204 FA07                DW      NULDVD          ;001 - RESERVED

                         IF      H17T
0206 F804                DW      H17DVD          ;010 - H17
                         ELSE
                         DW      NULDVD
                         ENDIF

                         IF      H37T
                         DW      H37DVD          ;011 - H37
                         ELSE
0208 FA07                DW      NULDVD
                         ENDIF

                         IF      H47T
                         DW      H47DVD          ;100 - H47
                         ELSE
020A FA07                DW      NULDVD
                         ENDIF

020C FA07                DW      NULDVD          ;101 - RESERVED

                         IF      H67T
                         DW      H67DVD          ;110 - H67
                         ELSE
020E FA07                DW      NULDVD
                         ENDIF

0210 FA07                DW      NULDVD          ;111 - RESERVED

                         PAGE
```

```
                        ;       SETSEC - SET SECTOR NUMBER
                        ;               ENTRY   C       DESIRED SECTOR (NUMBERED 1 TO SPT )
                        ;
0212 79         SETSEC: MOV     A,C             ;GET SECTOR NUMBER
0213 3D                 DCR     A               ;SAVE 0 TO SPT-1
0214 32200E             STA     SEKSEC
0217 C9                 RET

                        ;
                        ;       SETDMA - SET DISK I/O ADDRESS
                        ;

0218 60         SETDMA: MOV     H,B             ;MOVE ARGUMENT FROM BC TO HL
0219 69                 MOV     L,C
021A 222E0E             SHLD    DMAADR
021D C9                 RET

                        ;
                        ;       SECTRAN - TRANSLATE SECTOR INDEX USING TABLE AT DE
                        ;                 INTO SECTOR NUMBER FOR SKEW
                        ;
                        ;       ENTRY   C = SECTOR INDEX (0 TO SPT-1)
                        ;               DE = ADDR OF TRANSLATE TABLE
                        ;       EXIT    HL = SECTOR NUMBER (1 TO SPT)
                        ;

021E EB         SECTRAN: XCHG                   ;HL POINTS TO TABLE
021F 7C                 MOV     A,H             ;CHECK FOR NULL XLATE TABLE
0220 B5                 ORA     L
0221 09                 DAD     B
0222 CA2902             JZ      SECTRAN1        ;NULL XLATE TABLE
0225 6E                 MOV     L,M             ;L CONTAINS THE TRANSLATE SECTOR
0226 2600               MVI     H,0
0228 C9                 RET

                SECTRAN1:
0229 23                 INX     H               ;PUT IN RANGE 1 TO SPT
022A C9                 RET

                        PAGE
```

```
                        ;
                        ;       READ - READ THE (LOGICAL) RECORD SET BY SETDSK, SETTRK, SETSEC
                        ;               INTO MEMORY AT DMAADR, DEBLOCKING AS NECESSARY
                        ;
0000 =          WRALL   EQU     0               ;WRITE TO ALLOCATED
0001 =          WRDIR   EQU     1               ;WRITE TO DIRECTORY
0002 =          WRUAL   EQU     2               ;WRITE TO UNALLOCATED
                        ;
                        ;
                        ;       READ THE SELECTED CP/M SECTOR
022B AF         READ:   XRA     A
022C 32330C             STA     UNACNT
022F 3E01               MVI     A,1
0231 322C0E             STA     READOP          ;READ OPERATION
0234 322B0E             STA     RSFLAG          ;MUST READ DATA
0237 3E02               MVI     A,WRUAL
0239 322D0E             STA     WRTYPE          ;TREAT AS UNALLOC
023C C30903             JMP     RWOPER          ;TO PERFORM THE READ

                        PAGE
```

```
                        ;       WRITE - WRITE THE (LOGICAL) RECORD SET BY SETDSK, SETTRK, SETSEC
                        ;               FROM MEMORY AT DMAADR, BLOCKING AS NECESSARY
                        ;

023F AF         WRITE:  XRA     A               ;0 TO ACCUMULATOR
0240 322C0E             STA     READOP          ;NOT A READ OPERATION
0243 79                 MOV     A,C             ;WRITE TYPE IN C
0244 322D0E             STA     WRTYPE
0247 FE02               CPI     WRUAL           ;WRITE UNALLOCATED?
0249 C28D02             JNZ     CHKUNA          ;CHECK FOR UNALLOC
                        ;
                        ;       WRITE TO UNALLOCATED, SET PARAMETERS
024C 2A080E             LHLD    DPBX            ;SET NUMBER OF RECORDS PER ALLOCATION
024F 23                 INX     H
0250 23                 INX     H
0251 23                 INX     H
0252 7E                 MOV     A,M
0253 32330C             STA     UNACNT
0256 3A1D0E             LDA     SEKDSK          ;DISK TO SEEK
0259 32260E             STA     UNADSK          ;UNADSK = SEKDSK
025C 2A1E0E             LHLD    SEKTRK
025F 22270E             SHLD    UNATRK          ;UNATRK = SEKTRK

0262 2A080E             LHLD    DPBX
0265 11F0FF             LXI     D,-DPEHTH
0268 19                 DAD     D               ;HL POINTS TO POINTER TO XLATE TABLE
0269 CDB308             CALL    HLIHL           ;HL POINTS TO XLATE TABLE

026C 7C                 MOV     A,H
026D B5                 ORA     L
026E C27A02             JNZ     WRITE0          ;XLATE TABLE PRESENT
0271 3A200E             LDA     SEKSEC          ;NO XLATE TABLE
0274 32290E             STA     UNASI           ;  USE SECTOR #
0277 C38D02             JMP     CHKUNA

027A 3A200E     WRITE0: LDA     SEKSEC          ;GET DESIRED SECTOR
027D 3C                 INR     A               ;CORRECT TO 1 TO SPT
027E 0E00               MVI     C,0             ;INITIALIZE INDEX

0280 BE         WRITE1: CMP     M               ;FIND SECTOR'S INDEX
0281 CA8902             JZ      WRITE2          ;  (WHICH IS THE UNTRANSLATED SECTOR-1)

0284 0C                 INR     C               ;NOT THIS ONE, TRY THE NEXT
0285 23                 INX     H
0286 C38002             JMP     WRITE1

0289 79         WRITE2: MOV     A,C             ;GET THE INDEX
028A 32290E             STA     UNASI           ;SAVE IT

                        ;
                        ;       CHECK FOR WRITE TO UNALLOCATED SECTOR
028D 3A330C     CHKUNA: LDA     UNACNT          ;ANY UNALLOC REMAIN?
0290 B7                 ORA     A
0291 CA0103             JZ      ALLOC           ;SKIP IF NOT
                        ;
```

```
                        ;       MORE UNALLOCATED RECORDS REMAIN
0294 3D                         DCR     A               ;UNACNT = UNACNT-1
0295 32330C                     STA     UNACNT
0298 3A1D0E                     LDA     SEKDSK          ;SAME DISK?
029B 21260E                     LXI     H,UNADSK
029E BE                         CMP     M               ;SEKDSK = UNADSK?
029F C20103                     JNZ     ALLOC           ;SKIP IF NOT
                        ;
                        ;       DISKS ARE THE SAME
02A2 3A1E0E                     LDA     SEKTRK
02A5 21270E                     LXI     H,UNATRK
02A8 BE                         CMP     M               ;SEKTRK LSB = UNATRK LSB?
02A9 C20103                     JNZ     ALLOC           ;SKIP IF NOT
02AC 3A1F0E                     LDA     SEKTRK+1
02AF 23                         INX     H
02B0 BE                         CMP     M               ;SEKTRK MSB = UNATRK MSB?
02B1 C20103                     JNZ     ALLOC           ;SKIP IF NOT
                        ;
                        ;       TRACKS ARE THE SAME
02B4 2A080E                     LHLD    DPBX
02B7 11F0FF                     LXI     D,-DPEHTH
02BA 19                         DAD     D
02BB CDB308                     CALL    HLIHL
02BE 7C                         MOV     A,H
02BF B5                         ORA     L
02C0 C2CC02                     JNZ     CHKUNA5         ;XLATE TABLE PRESENT
02C3 3A290E                     LDA     UNASI
02C6 21200E                     LXI     H,SEKSEC
02C9 C3D602                     JMP     CHKUNA6
02CC 3A290E             CHKUNA5: LDA    UNASI
02CF CD9508                     CALL    DADA
02D2 3A200E                     LDA     SEKSEC
02D5 3C                         INR     A
02D6 BE                 CHKUNA6: CMP    M               ;SEKSEC = UNASI?
02D7 21290E                     LXI     H,UNASI
02DA C20103                     JNZ     ALLOC           ;SKIP IF NOT
                        ;
                        ;       MATCH, MOVE TO NEXT SECTOR FOR FUTURE REF
02DD 34                         INR     M               ;UNASI = UNASI+1
02DE 7E                         MOV     A,M             ;END OF TRACK?
02DF E5                         PUSH    H
02E0 F5                         PUSH    PSW
02E1 2A080E                     LHLD    DPBX            ;GET NUMBER OF SECTORS/TRACK FROM DPB
02E4 11FAFF                     LXI     D,-DPEHTH+DPEDPB
02E7 19                         DAD     D
02E8 CDB308                     CALL    HLIHL           ;GET DBPX
02EB F1                         POP     PSW
02EC BE                         CMP     M               ;FIRST ENTRY OF WHICH IS SEC/TRACK
02ED E1                         POP     H
02EE DAFA02                     JC      NOOVF           ;SKIP IF NO OVERFLOW
                        ;
                        ;       OVERFLOW TO NEXT TRACK
02F1 3600                       MVI     M,0             ;UNASI = 0
02F3 2A270E                     LHLD    UNATRK
02F6 23                         INX     H
02F7 22270E                     SHLD    UNATRK          ;UNATRK = UNATRK+1
```

```
                        ;
                        ;       MATCH FOUND, MARK AS UNNECESSARY READ
02FA AF         NOOVF:  XRA     A               ;0 TO ACCUMULATOR
02FB 322B0E             STA     RSFLAG          ;RSFLAG = 0
02FE C30903             JMP     RWOPER          ;TO PERFORM THE WRITE
                        ;
                        ;       NOT AN UNALLOCATED RECORD, REQUIRES PRE-READ
0301 AF         ALLOC:  XRA     A               ;0 TO ACCUM
0302 32330C             STA     UNACNT          ;UNACNT = 0
0305 3C                 INR     A               ;1 TO ACCUM
0306 322B0E             STA     RSFLAG          ;RSFLAG = 1

                        PAGE
```

```
                        ;       ENTER HERE TO PERFORM THE LOGICAL READ/WRITE
0309 AF         RWOPER: XRA     A               ;ZERO TO ACCUM
030A 322A0E             STA     ERFLAG          ;NO ERRORS (YET)
030D 2A080E             LHLD    DPBX            ;FIND LOGICAL SECTORS PER PHYSICAL
0310 4E                 MOV     C,M             ;GET DISK TYPE
0311 23                 INX     H
0312 23                 INX     H
0313 46                 MOV     B,M             ;GET LOGICAL SECTORS PER PHYSICAL

                        IF      H47T OR H67T
                        MOV     A,C
                        ANI     DPETYPE
                        CPI     DPEH47          ;CHECK FOR H47 FLOPPY
                        JZ      RWOX            ;IS H47
                        MOV     A,C
                        ANI     DPETYPF
                        CPI     DPEH67F         ;CHECK FOR H67 FLOPPY
                        JNZ     RWO             ;BR IF NOT
                RWOX:   LHLD    SEKTRK          ;WHAT TRACK ARE WE AFTER?
                        MOV     A,H
                        ORA     L               ;IF NOT TRACK 0
                        JNZ     RWO             ;  THEN USE THE VALUE OF LSP IN DPBX
                        MVI     B,1             ;  ELSE ON TRACK 0 IT IS 1!
                        ENDIF

0314 78         RWO:    MOV     A,B
0315 32180E             STA     LSP

0318 3A200E             LDA     SEKSEC          ;COMPUTE HOST SECTOR
031B F5         RW1:    PUSH    PSW             ;SAVE THE PHSYICAL SECTOR (TO DATE)
031C 78                 MOV     A,B             ;GET SHIFT FACTOR (SHIFT LOG2 SEC PER REC)
031D 1F                 RAR
031E 47                 MOV     B,A
031F DA2803             JC      RW2
0322 F1                 POP     PSW
0323 B7                 ORA     A
0324 1F                 RAR
0325 C31B03             JMP     RW1
0328 F1         RW2:    POP     PSW
0329 32250E             STA     SEKHST          ;HOST SECTOR TO SEEK
                        ;
                        ;       ACTIVE HOST SECTOR?
032C 21310C             LXI     H,HSTACT        ;HOST ACTIVE FLAG
032F 7E                 MOV     A,M
0330 3601               MVI     M,1             ;ALWAYS BECOMES 1
0332 B7                 ORA     A               ;WAS IT ALREADY?
0333 CA6303             JZ      FILHST          ;FILL HOST IF NOT
                        ;
                        ;       HOST BUFFER ACTIVE, SAME AS SEEK BUFFER?
0336 3A1D0E             LDA     SEKDSK
0339 21210E             LXI     H,HSTDSK        ;SAME DISK?
033C BE                 CMP     M               ;SEKDSK = HSTDSK?
033D C25C03             JNZ     NOMATCH
                        ;
                        ;       SAME DISK, SAME TRACK?
```

```
0340 3A1E0E            LDA     SEKTRK
0343 21220E            LXI     H,HSTTRK
0346 BE                CMP     M               ;SEKTRK LSB = HSTTRK LSB?
0347 C25C03            JNZ     NOMATCH
034A 3A1F0E            LDA     SEKTRK+1
034D 23                INX     H
034E BE                CMP     M               ;SEKTRK MSB = HSTTRK MSB?
034F C25C03            JNZ     NOMATCH
                ;
                ;       SAME DISK, SAME TRACK, SAME BUFFER?
0352 3A250E            LDA     SEKHST
0355 21240E            LXI     H,HSTSEC        ;SEKHST = HSTSEC?
0358 BE                CMP     M
0359 CA8E03            JZ      MATCH           ;SKIP IF MATCH
                ;
                ;       PROPER DISK, BUT NOT CORRECT SECTOR
035C 3A320C     NOMATCH: LDA   HSTWRT          ;HOST WRITTEN?
035F B7                ORA     A
0360 C4EF03            CNZ     WRITEHST        ;CLEAR HOST BUFF
                ;
                ;       MAY HAVE TO FILL THE HOST BUFFER
0363 3A1D0E     FILHST: LDA    SEKDSK
0366 32210E            STA     HSTDSK
0369 2A1E0E            LHLD    SEKTRK
036C 22220E            SHLD    HSTTRK
036F 3A250E            LDA     SEKHST
0372 32240E            STA     HSTSEC
0375 3A2C0E            LDA     READOP
0378 B7                ORA     A
0379 C28303            JNZ     FIL1            ;YES IT WAS A READ
037C 3A180E            LDA     LSP
037F 3D                DCR     A
0380 CA8A03            JZ      FIL2            ;DON'T NEED TO PREREAD IF PHYSICAL=LOGICAL

0383 3A2B0E     FIL1:  LDA     RSFLAG          ;NEED TO READ?
0386 B7                ORA     A
0387 C4DF03            CNZ     READHST         ;YES, IF 1
038A AF         FIL2:  XRA     A               ;0 TO ACCUM
038B 32320C            STA     HSTWRT          ;NO PENDING WRITE
                ;
                ;       COPY DATA TO OR FROM BUFFER
038E 3A180E     MATCH: LDA     LSP
0391 3D                DCR     A
0392 21200E            LXI     H,SEKSEC
0395 A6                ANA     M
0396 210000            LXI     H,0
0399 CAA403            JZ      M2
039C 118000            LXI     D,128
039F 19         M1:    DAD     D
03A0 3D                DCR     A
03A1 C29F03            JNZ     M1
                ;       HL HAS RELATIVE HOST BUFFER ADDRESS
03A4 11340C     M2:    LXI     D,HSTBUF
03A7 19                DAD     D               ;HL = HOST ADDRESS
03A8 EB                XCHG                    ;NOW IN DE
03A9 2A2B0E            LHLD    DMAADR          ;GET/PUT CP/M DATA
```

```
03AC 0E80                MVI     C,128           ;LENGTH OF MOVE
03AE 3A2C0E              LDA     READOP          ;WHICH WAY?
03B1 B7                  ORA     A
03B2 C2BB03              JNZ     RWMOVE          ;SKIP IF READ
                  ;
                  ;      WRITE OPERATION, MARK AND SWITCH DIRECTION
03B5 3E01                MVI     A,1
03B7 32320C              STA     HSTWRT          ;HSTWRT = 1
03BA EB                  XCHG                    ;SOURCE/DEST SWAP
                  ;
                  ;      C INITIALLY 128, DE IS SOURCE, HL IS DEST
03BB CDB808       RWMOVE: CALL   MOVEITX         ;MOVE DATA
                  ;
                  ;      DATA HAS BEEN MOVED TO/FROM HOST BUFFER
03BE 3A2D0E              LDA     WRTYPE          ;WRITE TYPE
03C1 FE01                CPI     WRDIR           ;TO DIRECTORY?
03C3 3A2A0E              LDA     ERFLAG          ;IN CASE OF ERRORS
03C6 C2D703              JNZ     RW9             ; BR IF NOT DIRECTORY WRITE
                  ;
                  ;      CLEAR HOST BUFFER FOR DIRECTORY WRITE
03C9 B7                  ORA     A               ;ERRORS?
03CA C2D903              JNZ     RW9A            ; BR IF ERROR
03CD AF                  XRA     A               ;0 TO ACCUM
03CE 32320C              STA     HSTWRT          ;BUFFER WRITTEN
03D1 CDEF03              CALL    WRITEHST
03D4 3A2A0E              LDA     ERFLAG

03D7 A7           RW9:   ANA     A               ;CHECK FOR ERROR
03D8 C8                  RZ                      ; RET IF NONE

03D9 F5           RW9A:  PUSH    PSW             ;SAVE ERROR
03DA CD0E04              CALL    FLUSH1          ;RE-INIT HOST BUFFER FLAGS
03DD F1                  POP     PSW             ;RESTORE ERROR
03DE C9                  RET

                         PAGE
```

```
                        ;       WRITEHST PERFORMS THE PHYSICAL WRITE TO
                        ;       THE HOST DISK, READHST READS THE PHYSICAL
                        ;       DISK.
                        ;
                        ;       READHST - PERFORM PHYSICAL SECTOR READ
                        ;               HSTDSK = HOST DISK #, HSTTRK = HOST TRACK #,
                        ;               HSTSEC = HOST SECT #. READ "HSTSIZ" BYTES
                        ;               INTO HSTBUF AND RETURN ERROR FLAG IN ERFLAG.
                        ;

                        READHST:
03DF AF                 XRA     A               ;INDICATE READ OPERATION
03E0 32170E             STA     RWOP

03E3 3A210E             LDA     HSTDSK          ;GET HOST DISK
03E6 CD4D04             CALL    SHD             ;SET HOST DEVICE POINTER
                                                ;  AND PERFORM LOGICAL TO PHYSICAL MAPPING

03E9 110300             LXI     D,DDRD
03EC C3F001             JMP     DSKDIS

                        ;
                        ;       WRITEHST - WRITE PHYSICAL SECTOR
                        ;               HSTDSK = HOST DISK #, HSTTRK = HOST TRACK #,
                        ;               HSTSEC = HOST SECT #, WRITE "HSTSIZ" BYTES
                        ;               FROM HSTBUF AND RETURN ERROR FLAG IN ERFLAG.
                        ;               RETURN ERFLAG NON-ZERO IF ERROR
                        ;

                        WRITEHST:
03EF 3E01               MVI     A,1             ;INDICATE WRITE OPERATION
03F1 32170E             STA     RWOP

03F4 3A210E             LDA     HSTDSK
03F7 CD4D04             CALL    SHD             ;GET HSTDPB, ALSO DO PHYSICAL -> LOGICAL MAP

03FA 110600             LXI     D,DDWR
03FD C3F001             JMP     DSKDIS

                        ;
                        ;       FLUSH - FLUSHES HOST BUFFER.
                        ;       FLUSH1 - REINITIALIZES HOST FLAGS
                        ;

0400 3A310C     FLUSH:  LDA     HSTACT          ;IS HOST BUFFERING ACTIVE
0403 A7                 ANA     A
0404 CA0E04             JZ      FLUSH1          ;BR IF NOT

0407 3A320C             LDA     HSTWRT          ;IS HOST BUFFER WAITING TO BE WRITTEN
040A A7                 ANA     A
040B C4EF03             CNZ     WRITEHST        ;YES - WRITE IT

040E AF         FLUSH1: XRA     A
040F 32310C             STA     HSTACT          ;DEACTIVATE HOST BUFFER
0412 32320C             STA     HSTWRT          ;NO PENDING WRITE
```

```
0415 32330C          STA     UNACNT          ;NO UNALLOCATED SECTORS IN BLOCK

0418 C9              RET

                ;  EXTENDED ERROR MESSAGE HANDLER.
                ;
                ;  ENTRY:  (ERRTYP) = ERROR CODE
                ;          (HL) = ADDR OF DEVICE TYPE MSG
                ;  EXIT:   NONE
                ;  USES:   ALL

0419 3A3600     PRTERR: LDA     MODE            ;CHECK IF EXTENDED ERROR MESSAGES
041C E602               ANI     MODEB1          ; REQUESTED
041E C8                 RZ                      ;  RET IF NOT

041F E5                 PUSH    H               ;SAVE DEVICE TYPE MSG
0420 21000C             LXI     H,CRLF
0423 CD9D0B             CALL    PMSG
0426 E1                 POP     H               ;RESTORE DEVICE TYPE MSG
0427 CD9D0B             CALL    PMSG

042A 3A170E             LDA     RWOP            ;PRINT I/O TYPE MSG
042D 21EB0B             LXI     H,RDMSG         ;ASSUME READ
0430 A7                 ANA     A
0431 CA3704             JZ      PRTERR1
0434 21F10B             LXI     H,WRMSG         ;WAS WRITE
                PRTERR1:
0437 CD9D0B             CALL    PMSG

043A 21F80B             LXI     H,ERRMSG
043D CD9D0B             CALL    PMSG

0440 3A1A0E             LDA     ERRTYP          ;PRINT ERROR TYPE
0443 CDAA0B             CALL    HOUT

0446 21000C             LXI     H,CRLF
0449 CD9D0B             CALL    PMSG

044C C9                 RET

                        PAGE
```

```
                        ; SHD - SET HOST DPB POINTER
                        ;       ENTRY   A       HOST DISK
                        ;       EXIT    HSTDPB  POINTS TO HOST DISK PARAMETERS
                        ;               (HL) = (HSTDPB)
                        ;

                        SHD:
                                IF      H17T OR H37T
044D 32CC04                     STA     SHDA
0450 32CD04                     STA     SHDB
                                ENDIF
0453 CDAB08                     CALL    GETDPEX         ;GET ADDR OF DPE'S HEATH EXTENSIONS
0456 220A0E                     SHLD    HSTDPB          ; SAVE IT

                                IF      H17T OR H37T

0459 7E                         MOV     A,M             ;CHECK DRIVE TYPE
045A E6E0                       ANI     DPETYPE
045C FE40                       CPI     DPEH17
045E CA6604                     JZ      SHD1
0461 FE60                       CPI     DPEH37
0463 C2C404                     JNZ     SHD6

0466 EB         SHD1:           XCHG                    ;(DE) = ADDR OF HEATH EXTENSIONS
0467 210600                     LXI     H,DPEFLG2-DPEHTH
046A 19                         DAD     D
046B 7E                         MOV     A,M
046C E602                       ANI     DPEIMG          ;CHECK IF IMAGINARY DRIVE
046E CA7D04                     JZ      SHD2            ; BR IF NOT

0471 210700                     LXI     H,DPELUN-DPEHTH ;GET REAL DRIVE LOGICAL UNIT #
0474 19                         DAD     D
0475 7E                         MOV     A,M
0476 32CD04                     STA     SHDB            ;SAVE IT
0479 CDAB08                     CALL    GETDPEX         ;GET ADDR OF DPE'S HEATH EXTENSIONS
047C EB                         XCHG                    ;(DE) = ADDR OF HEATH EXTENSIONS

047D 210700     SHD2:           LXI     H,DPELUN-DPEHTH
0480 19                         DAD     D
0481 3ACC04                     LDA     SHDA
0484 BE                         CMP     M               ;REQ UNIT = CURRENTLY MOUNTED UNIT
0485 CAC104                     JZ      SHD5            ; BR IF YES

0488 77                         MOV     M,A             ;UPDATE MOUNTED UNIT SLOT

0489 CDC504                     CALL    SHD9            ;GET LOGICAL DISK NAME
048C C641                       ADI     'A'
048E 32D904                     STA     MNMSGA

0491 3ACD04                     LDA     SHDB            ;GET PHYSICAL DISK NAME
0494 CDC504                     CALL    SHD9
0497 C641                       ADI     'A'
0499 32E404                     STA     MNMSGB

049C 110C00                     LXI     D,DDMNT         ;MOUNT DISK DRIVE
```

```
049F 2A0A0E            LHLD    HSTDPB
04A2 CDF001            CALL    DSKDIS

04A5 21CE04            LXI     H,MNMSG         ;PROMPT USER TO CHANGE DISK
04A8 CD9D0B            CALL    PMSG

04AB CDF308    SHD3:   CALL    CONIN           ;GET A CHARACTER FROM THE CONSOLE

04AE FE0D              CPI     CR              ;IF CHAR == CR
04B0 CABB04            JZ      SHD4            ; THEN GO AHEAD

04B3 0E07              MVI     C,BELL          ;  ELSE RING BELL
04B5 CD0109            CALL    CONOUT
04B8 C3AB04            JMP     SHD3            ;   AND WAIT FOR ANOTHER CHARACTER

04BB 21000C    SHD4:   LXI     H,CRLF
04BE CD9D0B            CALL    PMSG

04C1 2A0A0E    SHD5:   LHLD    HSTDPB

                       ENDIF

04C4 C9        SHD6:   RET

                       IF      H17T OR H37T

               ;  GET LOGICAL UNIT #
               ;
               ;    ENTRY:  (A) = LOGICAL AND MAPPED UNIT #'S
               ;                  BIT 7-4 = LOGICAL UNIT #
               ;                  BIT 3-0 = MAPPED UNIT #
               ;    EXIT:   (A) = LOGICAL UNIT #
               ;    USES:   A,F

04C5 1F        SHD9:   RAR
04C6 1F                RAR
04C7 1F                RAR
04C8 1F                RAR
04C9 E60F              ANI     0FH
04CB C9                RET

04CC           SHDA    DS      1               ;LOGICAL/MAPPED DRIVE # OF REQ
04CD           SHDB    DS      1               ;LOGICAL/MAPPED DRIVE # OF REAL

04CE 0D0A505554MNMSG   DB      CR,LF,'PUT DISK '
04D9 2E20494E20MNMSGA  DB      '. IN DRIVE '
04E4 2E3A20414EMNMSGB  DB      '.: AND PRESS RETURN',0
                       ENDIF

                       PAGE
```

```
                        IF      H17T

                ;       H17 DEVICE DRIVE                06 JUL 1981

                H17DVD:
04F8 C33606             JMP     RDYH17          ;SELECT ENTRY POINT
04FB C30705             JMP     RD17M           ;READ
04FE C32005             JMP     WR17M           ;WRITE
0501 C32906             JMP     RESH17          ;RESET
0504 C32006             JMP     MNTH17          ;MOUNT

                        PAGE
```

```
0507 3A240E   RD17M:  LDA   HSTSEC
050A 32150E           STA   SECTOR

050D 3A220E           LDA   HSTTRK        ;GET TRACK (# IS ALWAYS < 256)
0510 32140E           STA   TRACK

0513 CD4505           CALL  RD17
0516 3E00             MVI   A,00H
0518 D21C05           JNC   RDH1          ;CARRY SET INDICATES ERROR
051B 3D               DCR   A             ;  IF ERROR, MAKE ERFLAG = OFFH
051C 322A0E   RDH1:   STA   ERFLAG
051F C9               RET

0520 3A240E   WR17M:  LDA   HSTSEC        ;GET SECTOR NUMBER
0523 32150E           STA   SECTOR

0526 3A220E           LDA   HSTTRK        ;GET TRACK (# IS ALWAYS < 256)
0529 32140E           STA   TRACK

052C CD7A05           CALL  WR17
052F 3E00             MVI   A,00H
0531 D23505           JNC   WRH1          ;CARRY SET ON ERROR
0534 3D               DCR   A             ;  MAKE ERFLAG = OFFH ON ERROR
0535 322A0E   WRH1:   STA   ERFLAG
0538 C9               RET

                      PAGE
```

```
                        ;
                        ; XOK - EXIT FROM DISK OPERATION WITHOUT ERROR
                        ; XIT - EXIT FLAGGING ERROR IN CARRY
                        ;
0539 AF         XOK:    XRA     A               ;CLEAR CARRY
053A F5         XIT:    PUSH    PSW
053B F3                 DI
053C 210F06             LXI     H,DELAYS        ;SET DISK MOTOR AND SELECT TIMERS
053F 221808             SHLD    DLYMO
0542 F1                 POP     PSW
0543 FB                 EI
0544 C9                 RET
                        ;
                        ; RD17 - READ A SELECTED SECTOR
                        ;

0545 CDAF06     RD17:   CALL    SDP             ;SET PARAMETERS FOR THIS OPERATION

0548 CDCB06     RD171:  CALL    SDT             ;SEEK THE DESIRED TRACK

054B CD2C07             CALL    LPS             ;FIND THE PROPER SECTOR
054E DA7105             JC      RD17E           ;COULDN'T FIND IT
0551 0600               MVI     B,0             ;READ 256 BYTES
0553 21340C             LXI     H,HSTBUF        ;POINTER TO BUFFER
0556 CDB607             CALL    WSC             ;WAIT FOR SYNC
0559 3E20               MVI     A,D$E$MDS       ;MISSING DATA SYNC ERROR
055B DA7105             JC      RD17E           ;MISSING SYNC BYTE

055E CDA807     RD172:  CALL    RDB             ;READ A BYTE FROM THE DISK
0561 77                 MOV     M,A             ;PUT IT IN MEMORY
0562 23                 INX     H               ;INCREMENT POINTER

0563 05                 DCR     B               ;COUNT BYTE AS READ
0564 C25E05             JNZ     RD172           ;MORE TO READ

0567 42                 MOV     B,D
0568 CDA807             CALL    RDB             ;CHECK CHECKSUM
056B B8                 CMP     B
056C CA3905             JZ      XOK             ;EVERYTHING IS OKAY
056F 3E08               MVI     A,D$E$CHK       ;SIGNAL CHECKSUM ERROR

0571 CDC805     RD17E:  CALL    H17E            ;H17 ERROR HANDLER
0574 D24805             JNC     RD171           ;TRY AGAIN
0577 C33A05             JMP     XIT             ;RETURN, FLAGGING ERROR IN CARRY

                        PAGE
```

```
                        ;
                        ; WR17 - WRITE A SECTOR
                        ;

057A CDAF06     WR17:   CALL    SDP             ;SET DISK PARAMETERS

057D DB7F       WR171:  IN      DPDC            ;SEE IF WRITE PROTECTED
057F E604               ANI     DFWP
0581 3E40               MVI     A,D$E$WRP       ;POSSIBLE WRITE PROTECT ERROR
0583 C2BF05             JNZ     WR17E           ;YES, IT IS A WRITE PROTECT ERROR

0586 CDCB06             CALL    SDT             ;GET CORRECT TRACK

0589 CD2C07             CALL    LPS             ;FIND THE PROPER SECTOR
058C DABF05             JC      WR17E           ;COULDN'T FIND IT

058F 0600               MVI     B,0             ;256 BYTES/SECTOR
0591 21340C             LXI     H,HSTBUF        ;POINTER TO SOURCE OF DATA

0594 3E14               MVI     A,WRITA
0596 3D         WR172:  DCR     A
0597 C29605             JNZ     WR172

059A 0E0A               MVI     C,WRITB
059C 3E10               MVI     A,WRITC
059E CDD707             CALL    WSP             ;WRITE THE SYNC PATTERN

05A1 7E         WR173:  MOV     A,M
05A2 CDEC07             CALL    WNB             ;WRITE THIS DATA BYTE
05A5 23                 INX     H
05A6 05                 DCR     B
05A7 C2A105             JNZ     WR173           ;LOOP TO WRITE ALL 256 BYTES

05AA 7A                 MOV     A,D             ;WRITE CHECKSUM
05AB CDEC07             CALL    WNB

05AE CDEC07             CALL    WNB             ;CONTINUE TUNNEL ERASE
05B1 CDEC07             CALL    WNB             ;FOR 3 CHARACTER TIMES
05B4 CDEC07             CALL    WNB

05B7 3A0F00             LDA     DEVCTL          ;OFF WRITE GATE
05BA D37F               OUT     DPDC
05BC C33905             JMP     XOK

05BF CDC805     WR17E:  CALL    H17E            ;CALL THE H17 ERROR HANDLER
05C2 D27D05             JNC     WR171           ;TRY AGAIN
05C5 C33A05             JMP     XIT             ;RETURN FLAGGING ERROR IN CARRY

                        PAGE
```

```
                        ; H17E - H17 ERROR HANDLER
                        ;       ENTRY   A - ERROR TYPE
                        ;       EXIT    C - SET IF RETRIES EXHAUSTED

05C8 FB         H17E:   EI
05C9 321A0E             STA     ERRTYP          ;SAVE THE ERROR TYPE

05CC E6C0               ANI     D$E$UNR+D$E$WRP ;IS IT UNIT NOT READ OR
                                                ; WRITE PROTECT VIOLATION
05CE C21406             JNZ     H17E4           ;IF SO, THEN AUTOMATIC HARD ERROR

05D1 2A4900             LHLD    SECNT17         ;BUMP THE SOFT ERROR COUNT
05D4 23                 INX     H
05D5 224900             SHLD    SECNT17

05D8 21190E             LXI     H,ERRCNT        ;GET A POINTER TO THE RETRY COUNTER
05DB 35                 DCR     M               ;  DECREMENT THE RETRY COUNTER
05DC CA1406             JZ      H17E4           ;RETRIES EXHAUSTED, FLAG HARD ERROR

                        ;       BASED ON ERROR TYPE AND RETRY NUMBER
                        ;         SELECT FROM THE FOLLOWING RETRY ACTIONS
                        ;               SEEK TRACK 0    IF BAD TRACK ERROR OR RETRY 5
                        ;               JUST TRY AGAIN  IF RETRY ODD
                        ;               MOVE IN THEN OUT        IF RETRY & 2 = 0
                        ;               MOVE OUT THEN IN        IF RETRY & 2 = 1

05DF 3A1A0E             LDA     ERRTYP
05E2 FE01               CPI     D$E$TRK         ;WAS IT A BAD TRACK ERROR
05E4 CA0F06             JZ      H17E2           ;YES, GO DO A SEEK TRACK ZERO

05E7 7E                 MOV     A,M             ;FETCH RETRY COUNT
05E8 FE05               CPI     5               ;IF == 5
05EA CA0F06             JZ      H17E2           ;  THEN SEEK TRACK ZERO

05ED 1F                 RAR                     ;IF ODD, THEN TRY AGAIN IN PLACE
05EE 3F                 CMC                     ;COMPLEMENT CARRY
05EF D0                 RNC                     ;RETURN WITH CARRY CLEAR

05F0 2A1B0E             LHLD    TRKPT           ;GET POINTER TO CURRENT TRACK

05F3 1F                 RAR                     ;IS B1 OF ERRCNT = 1?
05F4 7E                 MOV     A,M             ;GET CURRENT TRACK
05F5 DA0406             JC      H17E1           ;YES, SO MOVE OUT THEN IN

05F8 FE27               CPI     39              ;IF IT IS ALREADY AT THE MAXIMUM TRACK
05FA CA1206             JZ      H17E3           ;  THEN DON'T DO ANYTHING

05FD 34                 INR     M               ;INCREMENT CURRENT TRACK
05FE CDFD06             CALL    MAI             ;  MOVE ARM IN ONE TRACK
0601 C31206             JMP     H17E3

0604 B7         H17E1:  ORA     A               ;IF IT IS ALREADY AT MINIMUM TRACK
0605 CA1206             JZ      H17E3           ;  THEN DON'T DO ANYTHING

0608 35                 DCR     M               ;DECREMENT CURRENT TRACK
```

```
0609 CD0207          CALL    MAO             ;  MOVE ARM OUT ONE TRACK
060C C31206          JMP     H17E3

060F CDEE06   H17E2: CALL    STZ

0612 AF       H17E3: XRA     A               ;CLEAR CARRY
0613 C9              RET

                     HARD ERROR HAS OCCURRED
                       (OPTIONALLY) PRINT ERROR MESSAGE
                       RETURN WITH CARRY SET

0614 211C06   H17E4: LXI     H,H17MSG
0617 CD1904          CALL    PRTERR          ;PRINT EXTENED ERROR MESSAGE

061A 37              STC                     ;RETURN FLAGGING HARD ERROR IN CARRY
061B C9              RET

061C 48313700 H17MSG DB      'H17',0

                     PAGE
```

```
                        ;
                        ;  MNTH17 -- MOUNT
                        ;

0620 2A0A0E     MNTH17: LHLD    HSTDPB          ;SET TRACK POINTER TO UNKNOWN
0623 110400             LXI     D,DPETRK-DPEHTH
0626 19                 DAD     D
0627 3680               MVI     M,DPEUNK

                        ;+      JMP     RESH17          ;RESET DRIVE


                        ;
                        ;  RESH17 -- ABORT OPERATION/RESET H17
                        ;

0629 AF         RESH17: XRA     A
062A D37F               OUT     DPDC
062C 320F00             STA     DEVCTL
062F 210000             LXI     H,0
0632 221808             SHLD    DLYMO
0635 C9                 RET


                        ;
                        ;  RDYH17 -- CHECK IF UNIT READY
                        ;            1. DRIVE IS AVAILABLE
                        ;            2. CORRECT TYPE OF MEDIA HAS BEEN INSERTED
                        ;
                        ;  ENTRY: NONE
                        ;  EXIT:  PSW/C = 0 IF READY
                        ;               = 1 IF NOT READY
                        ;  USES:  ALL
                        ;
                        ;  THE NUMBER OF HOLE TRANSITIONS IS COUNTED IN THE TIME IT TAKES
                        ;  FOR 1 REVOLUTION (200MS).  IF THE DISK INSERTED IS HARD SECTORED,
                        ;  THEN 11 HOLES PLUS/MINUS 1 (22+-2 TRANSITIONS) SHOULD BE COUNTED,
                        ;  10 SECTOR HOLES AND 1 INDEX HOLE.
                        ;

                        RDYH17:
0636 CD6E06             CALL    ONH17           ;TURN ON MOTOR AND SELECT DRIVE

0639 211A08             LXI     H,DLYW          ;WAIT UNTIL DRIVE IS UP TO SPEED
                        RDYH17A:
063C 7E                 MOV     A,M
063D A7                 ANA     A
063E C23C06             JNZ     RDYH17A

0641 210B00             LXI     H,TICCNT        ;GET TIME VALUE
0644 3E64               MVI     A,100
```

```
0646 86          ADD     M
0647 47          MOV     B,A          ;(B) = TIME VALUE

0648 0E00        MVI     C,0          ;(C) = HOLE COUNTER
064A 51          MOV     D,C          ;(D) = INIT HOLE STATUS TO NO HOLE

                 RDYH17B:
064B DB7F        IN      DPDC         ;GET HOLE STATUS
064D E601        ANI     DFHD
064F BA          CMP     D            ;HAS IT CHANGED
0650 CA5506      JZ      RDYH17C      ; BR IF NOT

0653 57          MOV     D,A          ;SAVE NEW STATUS
0654 0C          INR     C            ;COUNT TRANSITION

                 RDYH17C:
0655 78          MOV     A,B          ;CHECK IF TIME UP
0656 BE          CMP     M
0657 C24B06      JNZ     RDYH17B      ; BR IF NOT
065A 79          MOV     A,C          ;TIME UP -- CHECK # OF HOLES
065B FE14        CPI     10*2
065D DA6606      JC      RDYH17D      ;IF < 10 THEN ERROR
0660 FE19        CPI     12*2+1
0662 3F          CMC
0663 D26B06      JNC     RDYH17E      ;IF <= 12 THEN OK

                 RDYH17D:
0666 3E80        MVI     A,D$E$UNR    ;  OTHERWISE SAY UNIT NOT READY
0668 CDC805      CALL    H17E         ;REPORT ERROR

                 RDYH17E:
066B C33A05      JMP     XIT          ;RETURN VIA 'XIT'

                 PAGE
```

```
                        ;
                        ;  ONH17 -- TURNS ON MOTOR, SELECT DRIVE, AND SET SETTLE DELAY COUNTERS.
                        ;

066E FB         ONH17:  EI                      ;MAKE CERTAIN INTERRUPTS ARE ENABLED

066F 210000             LXI     H,0
0672 221808             SHLD    DLYMO

0675 2A0A0E             LHLD    HSTDPB          ;GET DEVICE SELECT CODE
0678 23                 INX     H
0679 7E                 MOV     A,M
067A F610               ORI     DFMO            ;TURN ON THE MOTOR
067C D37F               OUT     DPDC

067E 47                 MOV     B,A
067F 210F00             LXI     H,DEVCTL        ;WHAT WAS ITS STATE?
0682 7E                 MOV     A,M
0683 E610               ANI     DFMO            ;WAS THE MOTOR ON?
0685 C29F06             JNZ     ONH17A          ;YES, DON'T HAVE TO WAIT FOR IT

0688 E5                 PUSH    H
0689 2A0A0E             LHLD    HSTDPB          ;FIND OUT HOW FAST IT COMES UP
068C 110500             LXI     D,DPESEK-DPEHTH
068F 19                 DAD     D
0690 7E                 MOV     A,M
0691 E1                 POP     H
0692 17                 RAL
                        IF      DPEMO-10000000B
                ¢:      DPEMO NE 10000000B
                        ENDIF
0693 3EFA               MVI     A,SPD           ;UP TO SPEED IN SPD * 4 MS
0695 D2AA06             JNC     ONH17B
0698 1F                 RAR                     ;NEW DRIVES UP IN 1/4 TIME
0699 1F                 RAR
069A E63F               ANI     03FH
069C C3AA06             JMP     ONH17B

069F 7E         ONH17A: MOV     A,M
06A0 E60E               ANI     U0+U1+U2        ;CHECK THE AVAILABLE UNITS
06A2 A0                 ANA     B               ;WAS THIS UNIT SELECTED?
06A3 3E00               MVI     A,0
06A5 C2AA06             JNZ     ONH17B          ;THIS HEAD WAS ALREADY LOADED
06A8 3E14               MVI     A,HLTG          ;MUST WAIT FOR HEAD LOAD TIMING
06AA 321A08     ONH17B: STA     DLYW

06AD 70                 MOV     M,B
                ;MOV    A,B
                ;STA    DEVCTL

06AE C9                 RET

                        PAGE
```

```
                        ;
                        ; SDP - SET DEVICE PARAMETERS
                        ;      SET RETRY COUNT, MAKE SURE MOTOR IS ON AND DRIVE SELECTED
                        ;
                        SDP:
06AF CD6E06             CALL    ONH17           ;TURN ON MOTOR AND SELECT DRIVE

06B2 3E0A               MVI     A,RETRIES       ;SET TRY COUNTER
06B4 32190E             STA     ERRCNT

06B7 2A0A0E             LHLD    HSTDPB          ;GET THE CURRENT TRACK FOR THIS DISK
06BA 110400             LXI     D,DPETRK-DPEHTH ;OFFSET TO TRACK IN DISK TABLES
06BD 19                 DAD     D               ;GET ADDRESS OF TRACK FOR THIS DRIVE
06BE 221B0E             SHLD    TRKPT
06C1 7E                 MOV     A,M
06C2 17                 RAL                     ;IF MSB IS 0
06C3 D0                 RNC                     ;  THEN TRACK IS POINTED TO BY TRKPT
                        ;CALL   STZ             ;  ELSE HEAD POSITION IS UNKNOWN AND
                        ;RET                    ;     IS ZEROED
06C4 C3EE06             JMP     STZ

                        PAGE
```

```
                        ; SDT - SEEK DESIRED TRACK
                        ;       SEEK TO TRACK UPDATING *TRKPT
                        ;
06C7 34         SDT0:   INR     M
06C8 CDFD06             CALL    MAI

06CB 2A1B0E     SDT:    LHLD    TRKPT
06CE 3A140E             LDA     TRACK
06D1 BE                 CMP     M
06D2 CADF06             JZ      SDT1            ;AT DESIRED TRACK
06D5 F2C706             JP      SDT0            ;  MUST MOVE ARM IN
                                                ;  ELSE MUST MOVE ARM OUT
06D8 35                 DCR     M
06D9 CD0207             CALL    MAO

06DC C3CB06             JMP     SDT

06DF 3A1A08     SDT1:   LDA     DLYW            ;MAKE CERTAIN TO DELAY FOR HEAD SETTLE TIME
06E2 FE06               CPI     HST             ;IS WAIT ALREADY > HEAD SETTLE
06E4 D0                 RNC                     ;IF SO, RETURN
06E5 3E06               MVI     A,HST           ;  ELSE BE SURE TO DELAY FOR HEAD SETTLE
06E7 321A08             STA     DLYW
06EA C9                 RET

                        ;
                        ; STZ - SEEK TRACK 0
                        ;       CALLED DURING ERROR RECOVERY AND TO INITIALLY POSITION HEADS

06EB CD0207     STZ0:   CALL    MAO             ;MOVE THE ARM OUT
06EE DB7F       STZ:    IN      DPDC            ;CHECK THE TRACK ZERO SENSOR
06F0 E602               ANI     DFTO
06F2 CAEB06             JZ      STZ0            ;IF NOT SET, THEN MOVE OUT ANOTHER TRACK
06F5 2A1B0E             LHLD    TRKPT           ;ZERO TRACK INDICATOR FOR THIS DRIVE
06F8 3600               MVI     M,0
06FA C3DF06             JMP     SDT1            ;HEAD SETTLE DELAY IN CASE GOING TO 0

                        PAGE
```

```
                        ; MAI - MOVE ARM IN
                        ; MAO - MOVE ARM OUT
                        ;

06FD 3E20               MAI:    MVI     A,DFDI          ;SET DIRECTION
06FF C30307                     JMP     MAO1
0702 AF                 MAO:    XRA     A               ;SET DIRECTION
0703 E5                 MAO1:   PUSH    H
0704 67                         MOV     H,A
0705 3A0F00                     LDA     DEVCTL          ;GET CURRENT VALUE OF DISK PORT
0708 B4                         ORA     H               ;OR IN DIRECTION
0709 D37F                       OUT     DPDC            ; SEND IT TO DISK
070B F640                       ORI     DFST            ;OR IN STEP
070D D37F                       OUT     DPDC            ; SEND IT TO DISK
070F EE40                       XRI     DFST            ;CLEAR STEP
0711 D37F                       OUT     DPDC            ; SEND IT TO DISK
0713 2A0A0E                     LHLD    HSTDPB          ;GET STEP RATE
0716 110500                     LXI     D,5
0719 19                         DAD     D
071A 7E                         MOV     A,M
071B E67F                       ANI     07FH
071D E1                         POP     H
                        ;       CALL    DLY             ;IMPLICIT CALL DLY AND RET
                        ;       RET

                        ;
                        ; DLY - DELAY A * 2 MS
                        ;

071E E5                 DLY:    PUSH    H
071F 210B00                     LXI     H,TICCNT        ;POINTER TO TICK COUNTER, INCREMENTED EVERY 2MS
0722 86                         ADD     M               ;VALUE OF TICCNT WHEN DELAY COMPLETED
0723 BE                 DLY1:   CMP     M               ;WAIT FOR TICCNT TO CATCH UP
0724 C22307                     JNZ     DLY1
0727 E1                         POP     H
0728 C9                         RET

                                PAGE
```

```
                        ; LPS - LOCATE PROPER SECTOR
                        ;

0729 CD6A07   LPS0:     CALL    STS             ;SKIP THIS SECTOR

072C 3A1A08   LPS:      LDA     DLYW            ;READY TO READ YET?
072F B7                 ORA     A
0730 C22907             JNZ     LPS0            ;IF NOT, WAIT A SECTOR TIME

0733 0614               MVI     B,LPSA

0735 F3       LPS1:     DI
0736 CDB607             CALL    WSC             ;WAIT FOR A SYNC CHARACTER
0739 3E02               MVI     A,D$E$HSY       ;FLAG HEADER SYNC ERROR
073B DA5F07             JC      LPS2            ;COULDN'T FIND ONE

073E CDA807             CALL    RDB             ;READ THE VOLUME NUMBER
0741 CDA807             CALL    RDB             ;READ THE TRACK NUMBER
0744 21140E             LXI     H,TRACK
0747 BE                 CMP     M
0748 3E01               MVI     A,D$E$TRK       ;BAD TRACK ERROR
074A C25F07             JNZ     LPS2            ;WRONG TRACK

074D CDA807             CALL    RDB             ;READ THE SECTOR NUMBER
0750 23                 INX     H               ;POINT TO SECTOR
0751 BE                 CMP     M
0752 3E10               MVI     A,D$E$RNF       ;RECORD NOT FOUND ERROR
0754 C25F07             JNZ     LPS2            ;WRONG SECTOR

0757 62                 MOV     H,D
0758 CDA807             CALL    RDB             ;DO CHECKSUM ON HEADER
075B BC                 CMP     H
075C C8                 RZ                      ;OKAY
075D 3E04               MVI     A,D$E$HCK       ;HEADER CHECKSUM IS WRONG

075F F5       LPS2:     PUSH    PSW
0760 CD6A07             CALL    STS             ;SKIP THIS SECTOR
0763 F1                 POP     PSW
0764 05                 DCR     B               ;ANOTHER TIME PASSES QUICKLY PAST
0765 C23507             JNZ     LPS1

0768 37                 STC                     ;ENOUGH ALREADY
0769 C9                 RET


                        PAGE
```

```
                        ; STS - SKIP THIS SECTOR
                        ;       EXIT AT BEGINNING OF NEXT SECTOR
                        ;       1. IF HEAD IS NOT OVER A HOLE, WAIT 8 MS WHILE HOLE CHECKING.
                        ;          IF NO HOLE IN THIS TIME, THEN WE ARE IN A REGULAR GAP.
                        ;          WAIT FOR THE NEXT HOLE AND EXIT.
                        ;       2. IF THE HEAD IS OVER A HOLE, OR BECOMES SO DURING THE 8 MS
                        ;          WAIT, THEN WAIT FOR THE HOLE TO PASS, WAIT 12 MS IN CASE OF
                        ;          THE INDEX HOLE, THEN WAIT FOR THE NEXT HOLE AND EXIT.
                        ;

076A FB         STS:    EI
076B C5                 PUSH    B
076C DB7F               IN      DPDC            ;CHECK THE DISK PORT
076E 1F                 RAR                     ;FOR SECTOR HOLES
076F DA8607             JC      STS2            ;CURRENTLY OVER A HOLE

                        ;       NO HOLE YET, WAIT 8 MS MIN (10 MS MAX) FOR HOLE TO APPEAR

0772 210B00             LXI     H,TICCNT
0775 46                 MOV     B,M
0776 DB7F       STS1:   IN      DPDC
0778 1F                 RAR
0779 DA8607             JC      STS2            ;FOUND A HOLE

077C 3E05               MVI     A,STSA
077E 80                 ADD     B
077F BE                 CMP     M
0780 C27607             JNZ     STS1            ;8 MS STILL NOT UP
0783 C38E07             JMP     STS3            ;FOUND A SECTOR GAP

                        ;       HAVE HOLE, SKIP IT AND WAIT 12 MS

0786 CD9B07     STS2:   CALL    WNH             ;WAIT FOR NO HOLE
0789 3E07               MVI     A,STSB
078B CD1E07             CALL    DLY
078E C1         STS3:   POP     B
078F F3                 DI

                        PAGE
```

```
                        ; WHD - WAIT HOLE DETECT
                        ;
0790 DB7F       WHD:    IN      DPDC            ;WATCH THE DISK CONTROL PORT
0792 1F                 RAR                     ;UNTIL A HOLE IS FOUND
0793 D29007             JNC     WHD             ; STILL NO HOLE
0796 3E14               MVI     A,WHDA          ;SET UP LOOP DELAY COUNT
0798 C3A307             JMP     UDLY

                        ;
                        ; WNH - WAIT FOR NO HOLE
                        ;
079B DB7F       WNH:    IN      DPDC            ;WATCH THE DISK CONTROL PORT
079D 1F                 RAR                     ;UNTIL THE CURRENT HOLE IS PAST
079E DA9B07             JC      WNH
07A1 3E14               MVI     A,WHNA          ;SET UP LOOP DELAY COUNT

                        ;
                        ; UDLY - MICROSECOND DELAY
                        ;       CALLED WITH INTERRUPTS DISABLED TO WAIT
                        ;       A * ( 15 / 2.048 ) MICROSECONDS ON 8080
                        ;       A * ( 14 / 2.048 ) MICROSECONDS ON Z80

07A3 3D         UDLY:   DCR     A
07A4 C2A307             JNZ     UDLY
07A7 C9                 RET

                        ;
                        ; RDB - READ BYTE FROM DISK
                        ;
07A8 DB7D       RDB:    IN      UPST            ;IS A BYTE READY?
07AA 1F                 RAR
07AB D2A807             JNC     RDB             ;  WAIT UNTIL READY

07AE DB7C               IN      UPDP            ;GET THE BYTE
07B0 5F                 MOV     E,A             ;SAVE IT IN E
07B1 AA                 XRA     D               ;UPDATE CRC
07B2 07                 RLC
07B3 57                 MOV     D,A
07B4 7B                 MOV     A,E             ;RESTORE BYTE READ TO A
07B5 C9                 RET

                        PAGE
```

```
                        ; WSC - WAIT SYNC CHARACTER
                        ;       WSC WAITS FOR THE APPEARANCE OF A SYNC CHARACTER.  THE DISK
                        ;       SHOULD BE SELECTED, MOVING, AND THE HEAD SHOULD BE OVER THE
                        ;       PRE-SYNC ZERO BAND
                        ;
                        ;       IF SYNC IS NOT FOUND IN 25 CHARACTER TIMES, ERROR

07B6 3E30               WSC:    MVI     A,READA         ;DELAY PAST GARBAGE BYTE
07B8 3D                 WSC0:   DCR     A
07B9 C2B807                     JNZ     WSC0
07BC 3EFD                       MVI     A,DSYN          ;SET UP SYNC CHARACTER
07BE D37E                       OUT     UPSC
07C0 DB7E                       IN      UPSR            ;RESET SYNC SEARCH
07C2 3E50                       MVI     A,WSCA          ;NUMBER OF LOOPS IN 25 CHARACTERS
07C4 57                         MOV     D,A
07C5 DB7F               WSC1:   IN      DPDC
07C7 E608                       ANI     DFSD            ;CHECK FOR SYNC
07C9 C2D207                     JNZ     WSC2            ;GOT IT
07CC 15                         DCR     D
07CD C2C507                     JNZ     WSC1            ;TRY UNTIL TIME-OUT

07D0 37                         STC                     ;COULDN'T FIND SYNC
07D1 C9                         RET

                        ;       FOUND SYNC CHARACTER

07D2 DB7C               WSC2:   IN      UPDP            ;GOBBLE THE SYNC CHARACTER
07D4 1600                       MVI     D,0             ;CLEAR CHECKSUM
07D6 C9                         RET

                                PAGE
```

```
                        ; WSP - WRITE SYNC PATTERN
                        ;       WSP WRITES A SYNC PATTERN OF ZEROS, FOLLOWED BY A SYNC CHARACTER
                        ;
                        ;       ENTRY   A       INITIAL DELAY COUNTER
                        ;               C       NUMBER OF ZERO BYTES TO WRITE

07D7 3D         WSP:    DCR     A               ;DELAY
07D8 C2D707             JNZ     WSP

                        ;       DELAY IS UP, TURN ON WRITE GATE

07DB 3A0F00             LDA     DEVCTL
07DE 3C                 INR     A               ;SET WRITE GATE ON
07DF D37F               OUT     DPDC

                        ;       WRITE # OF ZEROS SPECIFIED IN C

07E1 AF         WSP1:   XRA     A
07E2 CDEC07             CALL    WNB             ;WRITE A ZERO
07E5 0D                 DCR     C               ;COUNT IT
07E6 C2E107             JNZ     WSP1

07E9 3EFD               MVI     A,DSYN          ;WRITE A SYNC CHARACTER
07EB 57                 MOV     D,A             ;PRE-CLEAR CHECKSUM, SO WNB EXITS WITH D = 0
                        ;       JMP     WNB             ;IMPLICIT CALL, RETURN WNB

                        PAGE
```

```
                        ; WNB - WRITE NEXT BYTE
                        ;       WRITE A BYTE TO DISK PRESUMING WRITE GATE ALREADY ON
                        ;

07EC 5F         WNB:    MOV     E,A             ;SAVE CHARACTER TO BE WRITTEN
07ED DB7D       WNB1:   IN      UPST            ;IS USRT READY FOR ANOTHER CHARACTER
07EF A7                 ANA     A               ;SET FLAGS
07F0 F2ED07             JP      WNB1            ;NOT READY, WAIT SOME MORE

07F3 7B                 MOV     A,E             ;GET CHARACTER
07F4 D37C               OUT     UPDP            ;WRITE IT TO DISK
07F6 AA                 XRA     D               ;UPDATE CRC
07F7 07                 RLC
07F8 57                 MOV     D,A
07F9 C9                 RET

                        ENDIF

                        PAGE
```

```
                IF      H37T

;               H37 DISK DEVICE DRIVER                          27 AUG 81

H37DVD: JMP     SET37           ;SETDSK ENTRY POINT
        JMP     RD37            ;READ SECTOR
        JMP     WR37            ;WRITE SECTOR
        JMP     RESH37          ;RESET
        JMP     MNTH37          ;MOUNT

;
;  SELECT DISK 1ST LOGIN
;

SET37:
        LXI     H,0             ;TRACK 0
        SHLD    HSTTRK
        XRA     A               ;1ST SECTOR
        STA     HSTSEC          ; (SECTOR # 0 TO SPT-1)
        DCR     A               ;SET FLAG TO SUSPEND 'H37DONE'
        STA     SET37A          ; PROCESSING AND DO NO RETRIES

;  CHECK IF DRIVE IS AVAILABLE AND PROPER MEDIA INSERTED.

        CALL    RDYH37          ;CHECK IF DRIVE IS READY
        JC      SET379          ; BR IF NOT

;  READ LABEL.

        LHLD    DPBX            ;FORCE RESTORE OF HEAD
        LXI     D,DPETRK-DPEHTH
        DAD     D
        MVI     M,DPEUNK

        IF      H37ED

        CALL    RD37            ;TRY READING LABEL AT DENSITY
                                ; CURRENTLY INDICATED IN TABLES
        JZ      SET373          ;  BR IF SUCCESSFUL

        LHLD    DPBX            ;UNABLE TO READ LABEL
        MOV     A,M             ; TRY OTHER DENSITY
        XRI     DPEDD
        MOV     M,A
        CALL    RD37            ;TRY TO READ LABEL
        JNZ     SET379          ; BR IF ERROR

SET373:

        ELSE

        MVI     A,FDCRDA        ;TRY READ ADDRESS COMMAND AT DENSITY
        LXI     D,H37TMP        ; CURRENTLY INDICATED IN TABLES
        CALL    H37RD
        JZ      SET373          ;  BR IF SUCCESSFUL
```

```
        LHLD    DPBX            ;UNABLE TO READ
        MOV     A,M             ; CHANGE TO OTHER DENSITY
        XRI     DPEDD
        MOV     M,A
        LXI     H,H37CTL
        MOV     A,M
        XRI     CONMFM
        MOV     M,A
        OUT     FD$CON
        MVI     A,FDCRDA        ;TRY READ ADDRESS COMMAND AGAIN
        LXI     D,H37TMP
        CALL    H37RD
        JNZ     SET379          ; BR IF ERROR

SET373: LDA     H37TMP+FDRASL   ;Q. SECTOR LENGTH = 256
        CPI     FDSL256
        JNZ     SET379          ; BR IF NOT

        CALL    RD37            ;READ LABEL
        JNZ     SET379          ; BR IF ERROR

        ENDIF

        CALL    CHKLAB          ;CHECK CHECKSUM OF LABEL
        JZ      SET373A         ; BR IF CORRECT CHECKSUM

        LHLD    DPBX            ;IF SINGLE DENSITY THEN ASSUME NO LABEL
        MOV     A,M             ; IS PRESENT AND USE DEFAULT LABEL,
        ANI     DPEDD           ;  OTHERWISE ERROR
        JNZ     SET379
        MVI     C,LABLEN-1
        LXI     D,HSTBUF+LABEL
        LXI     H,DFTL37
        CALL    MOVEIT

;  MOVE LABEL INFO TO DPE'S HEATH EXTENSIONS.

SET373A:
        LHLD    DPBX            ;DENSITY/SIDES
        MOV     A,M
        ANI     DPETYPE+DPE96T
        MOV     B,A
        LXI     D,HSTBUF+LABHTH+DPEFLAG-DPEHTH
        LDAX    D
        ANI     DPEED+DPEDD+DPE2S
        ORA     B
        MOV     M,A

        INX     D               ;CP/M RECORDS PER SECTOR
        INX     D
        LDAX    D
        INX     H
        INX     H
        MOV     M,A
```

```
                INX     D               ;CP/M RECORDS PER ALLOCATION BLOCK
                LDAX    D
                INX     H
                MOV     M,A

                INX     D               ;DPEFLG2
                INX     D
                INX     D
                LDAX    D
                ANI     OFFH-DPEIMG
                MOV     B,A
                INX     H
                INX     H
                INX     H
                MOV     A,M
                ANI     DPEIMG
                ORA     B
                MOV     M,A
                IF      (DPEFLG2-DPERPAB-3)
%:              (DPEFLG2-DPERPAB) NE 3
                ENDIF

        ;  IF LABEL INDICATES THAT THE MEDIA IS DOUBLE SIDED, THEN
        ;  CHECK OUT THE DRIVE FOR DOUBLE SIDED CAPABILITY.

                LDA     HSTBUF+LABHTH+DPEFLAG-DPEHTH
                ANI     DPE2S
                JZ      SET374          ;BR IF MEDIA IS NOT DOUBLE SIDED

                MVI     A,FDCRDA+FDFSS1
                LXI     D,H37TMP
                CALL    H37RD           ;TRY TO READ A SECTOR HEADER ON 2ND SIDE
                JNZ     SET379          ; BR IF ERROR

                LDA     H37TMP+FDRASID  ;CHECK SIDE INFO
                CPI     1
                JNZ     SET379          ; BR IF NOT 2ND SIDE

        ;  STEP IN 2 TRACKS, READ ADDRESS, AND STEP BACK OUT 2 TRACKS.
        ;  IF READ ADDRESS FINDS TRACK 1 INSTEAD OF TRACK 2,
        ;  THEN ASSUME 48 TPI MEDIA GENERATED
        ;  ON A 48 TPI DRIVE WAS INSERTED INTO THE 96 TPI DRIVE; THEREFORE,
        ;  USE AS A R/O DISK WITH DOUBLE STEPPING.
        ;
        ;  THERE ARE TWO 48 TPI MEDIA FORMATS THAT ARE SUPPORTED ON
        ;  A 96 TPI DRIVE.
        ;     1) MEDIA WAS FORMATTED ON A 96 TPI DRIVE AND ONLY USES
        ;        THE FIRST HALF OF THE DISK SURFACE.  (E.G. A 48 TPI MEDIA
        ;        IS DUPLICATED ONTO MEDIA IN A 96 TPI DRIVE)
        ;     2) MEDIA WAS FORMATTED ON A 48 TPI DRIVE.  THE HARDWARE
        ;        GROUP HAS INFORMED ME THAT THE 96 TPI DRIVE CAN RELIABLY
        ;        READ SUCH MEDIA BUT CANNOT WRITE ON IT.  TO GO BETWEEN
        ;        TRACKS IT IS NECESSARY TO DOUBLE THE NUMBER OF STEPS.
        ;
        ;  THE FOLLOWING ARE THE POSSIBLE OUTCOMES OF THE READ ADDRESS
        ;     MEDIA         DRIVE          DRIVE        OUTCOME
```

```
;   FORMAT    GENERATED ON    INSERTED IN    TRACK
;   ------    ------------    -----------    -------
;   48 TPI      48 TPI          48 TPI         2
;   48 TPI      48 TPI          96 TPI         1
;   48 TPI      96 TPI          48 TPI       ERROR
;   48 TPI      96 TPI          96 TPI         2
;   96 TPI      96 TPI          48 TPI       ERROR
;   96 TPI      96 TPI          96 TPI         2
;

SET374:
        MVI     A,2
        STA     TRACK
        CALL    SDT37           ;STEP IN TWO TRACKS

        MVI     A,FDCRDA
        LXI     D,H37TMP
        CALL    H37RD           ;DO READ ADDRESS
        JNZ     SET379          ; BR IF ERROR

        LDA     H37TMP+FDRATRK  ;CHECK IF TRACK 2
        CPI     2
        JZ      SET374D         ; BR IF YES
                                ;  1) 48 TPI MEDIA GENERATED ON 48 TPI
                                ;     DRIVE INSERTED INTO 48 TPI DRIVE
                                ;  2) 48 TPI MEDIA GENERATED ON 96 TPI
                                ;     DRIVE INSERTED INTO 96 TPI DRIVE
                                ;  3) 96 TPI MEDIA INSERTED IN 96 TPI

        CPI     1               ;CHECK IF TRACK 1
        JNZ     SET379          ; BR IF NOT

        LHLD    DPBX            ;SET FLAG TO INDICATE 48 TPI MEDIA
        MOV     A,M             ; GENERATED ON 48 TPI DRIVE INSERTED
        ORI     DPE48RO         ;  INTO 96 TPI DRIVE.  THE MEDIA
        MOV     M,A             ;   IS TREATED AS R/O

SET374D: CALL   RST37           ;STEP OUT TWO TRACKS BY DOING RESTORE
                                ; ('TRKPT' VARIABLE WAS SET ABOVE)

;  MOVE LABEL INFO TO DISK PARAMETER BLOCK.

SET375: LHLD    SETDSKC         ;GET DPE ADDRESS
        LXI     B,DPEDPB        ;GET DPB ADDRESS
        DAD     B
        CALL    HLIHL
        LXI     D,HSTBUF+LABDPB ;GET ADDR OF INFO IN LABEL
        MVI     C,DPBL          ;COUNT TO MOVE
        CALL    MOVEITX         ;MOVE INFO

;  RETURN WITH DPE ADDRESS IN (HL).

        XRA     A               ;INDICATE NO ERROR

SET378: MVI     A,0             ;CLEAR 'H37DONE' SUSPENSION FLAG
        STA     SET37A          ; (MVI USED TO CLEAR ACCUMULATOR
```

```
                                        ;  SINCE I WANT TO PRESERVE PSW/C BIT)
        CALL    H37DONE                 ;DONE WITH DRIVE FOR NOW ('H37DONE'
                                        ; DOESN'T DISTURB PSW/C BIT EITHER)
        RET

;  ERROR OCCURRED.

SET379:
        STC                             ;INDICATE ERROR
        JMP     SET378

;

DFTL37  DB      LABVER                  ;DEFAULT LABEL
        DB      DPEH37,0,2,8,0,0,0,0
        DW      20
        DB      3,7,0
        DW      91,63,00C0H,16,3

SET37A  DB      0                       ;SUSPEND 'H37DONE' FLAG
                                        ; 0 = NO   NOT 0 = YES

H37TMP  DS      FDRAL

        PAGE
;
;  H37WAIT -- LOW LEVEL I/O ROUTINE TO ISSUE COMMAND AND WAIT
;             FOR COMPLETION.
;
;    ENTRY:  (A)=COMMAND
;            ((SP))=RET ADDR
;    EXIT:   (A)=STATUS BYTE
;            RET IS DONE VIA H37 INTERRUPT HANDLER
;    USES:  A,F,B,H,L
;

H37WAIT:
        POP     H                       ;GET RET ADDR
        SHLD    H37IRET                 ;INFORM INTERRUPT HANDLER

        MOV     B,A
        MVI     A,FD$CD                 ;ACCESS C/D REGS
        OUT     FD$INT
        MOV     A,B

        OUT     FD$CMD                  ;ISSUE COMMAND
        JMP     $                       ;WAIT


;
;  H37RD -- LOW LEVEL I/O ROUTINE TO INPUT DATA FROM H37.
;
;    ENTRY:  (A)=COMMAND
;            (DE)=BUFFER ADDRESS
```

```
;           ((SP))=RET ADDRESS
;     EXIT:   (A)=STATUS BYTE
;             PSW/Z = 0 IF ERROR
;                   = 1 IF NO ERROR
;     USES:  A,F,B,D,E,H,L
;

H37RD:
        LXI     H,H37RD2        ;SET INTERRUPT HANDLER RET ADDR
        SHLD    H37IRET

        MOV     B,A
        LDA     H37CTL          ;TURN ON DRQ
        ORI     CONDRQ
        OUT     FD$CON

        MVI     A,FD$CD         ;ACCESS C/D REGS
        OUT     FD$INT

        MOV     A,B
        LXI     H,H37RD1
        OUT     FD$CMD          ;ISSUE COMMAND

H37RD1:
        HLT                     ;WAIT FOR NEXT INPUT BYTE
        IN      FD$DAT          ;INPUT BYTE
        STAX    D               ;PUT IT INTO BUFFER
        INX     D               ;BUMP BUFFER POINTER
        PCHL                    ;LOOP

;  TURN OFF DRQ ROUTINE.  (ALSO USED BY WR37)
H37RD2:
        PUSH    PSW
        LDA     H37CTL          ;TURN OFF DRQ
        OUT     FD$CON
        POP     PSW

        ANA     A               ;SET PSW/Z TO INDICATE ERROR STATUS
        RET


;
;  H37DONE -- DONE WITH H37 FOR NOW
;
;     USES:  H,L
;
;     THE SELECT DISK DEVICE DRIVER ENTRY USES OTHER ROUTINES
;     IN THE DRIVER WHICH USE 'H37DONE'.  SINCE SELECT USES
;     LOW LEVEL I/O ROUTINES ALSO, THE 'H37DONE' PROCESSING
;     MUST BE TEMPORARILY SUSPENDED UNTIL SELECT IS DONE.
;

H37DONE:
        PUSH    PSW
```

```
        LDA     SET37A          ;CHECK IF 'H37DONE' IS SUSPENDED
        ANA     A               ; TEMPORARILY
        JNZ     H37DONE1        ; BR IF IT IS

        LXI     H,H37CTL        ;TURN OFF IRQ
        MOV     A,M
        ANI     OFFH-CONIRQ
        OUT     FD$CON
        MOV     M,A

        LXI     H,DELAY37       ;SET DESELECT AND MOTOR TURN OFF
        SHLD    DLYMO37         ; DELAY VALUES

H37DONE1:
        POP     PSW
        RET

        PAGE
;
;  RD37 -- HIGH LEVEL I/O ROUTINE TO READ H37 SECTORS.
;
;    ENTRY:  (HSTDPB) = ADDR OF HEATH EXTENSIONS FOR DRIVE
;            (HSTTRK) = TRACK
;            (HSTSEC) = SECTOR (0 TO SPT-1)
;    EXIT:   (ERFLAG) = 0 IF NO ERROR
;                     <> 0 IF ERROR
;            PSW/Z = 0 IF ERROR
;                  = 1 IF NO ERROR
;    USES:   ALL
;
RD37:   CALL    SDP37           ;SET UP

RD370:  CALL    SDT37           ;SEEK THE DESIRED TRACK

        LDA     SIDE            ;GET SIDE SELECT
        ORI     FDCRDS+FDFSLF   ;SET UP COMMAND
        LXI     D,HSTBUF        ;GET BUFFER ADDR
        CALL    H37RD           ;READ IN DATA
        JZ      RD373           ; BR IF NO ERRORS

        CALL    H37E            ;CHECK IF I SHOULD RETRY
        JNC     RD370           ; BR IF YES

        ORI     OFFH            ;  OTHERWISE FLAG ERROR
        STA     ERFLAG

RD373:  JMP     H37DONE         ;EXIT THRU 'H37DONE'

        PAGE
;
;  WR37 -- I/O ROUTINE TO WRITE SECTOR TO H37
;
;    ENTRY:  (HSTDPB) = ADDR OF HEATH EXTENSIONS FOR DRIVE
;            (HSTTRK) = TRACK
```

```
;           (HSTSEC) = SECTOR (0 TO SPT-1)
;    EXIT:  (ERFLAG) = 0 IF NO ERROR
;                    <> 0 IF ERROR
;           PSW/Z = 0 IF ERROR
;                 = 1 IF NO ERROR
;    USES:  ALL
;

WR37:
        LHLD    HSTDPB          ;CHECK FOR 48 TPI R/O
        MOV     A,M
        ANI     DPE48RO
        MVI     A,FDSWPV        ;ASSUME ERROR CODE
        JNZ     WR37E           ; BR IF IT IS -- NO WRITY

        CALL    SDP37           ;SET UP

WR370:  CALL    SDT37           ;SEEK THE DESIRED TRACK

        MVI     A,FD$CD
        OUT     FD$INT          ;ACCESS C/D REGS
        LDA     H37CTL
        ORI     CONDRQ          ;TURN ON DRQ
        OUT     FD$CON
        LDA     SIDE            ;GET SIDE
        ORI     FDCWRS+FDFSLF   ;SET UP COMMAND
        LXI     H,WR372         ;INTERRUPT RETURN ADDRESS
        SHLD    H37IRET
        LXI     H,WR371         ;LOOP ADDRESS
        LXI     D,HSTBUF
        OUT     FD$CMD          ;SEND THE COMMAND

WR371:  LDAX    D
        HLT
        OUT     FD$DAT
        INX     D
        PCHL

WR372:  CALL    H37RD2          ;TURN OFF DRQ USING 'H37RD2'

        JZ      WR373           ;BR IF NO I/O ERROR

WR37E:  CALL    H37E            ;CHECK IF I SHOULD RETRY
        JNC     WR370           ; BR IF YES

        ORI     OFFH            ;FLAG AS I/O ERROR
        STA     ERFLAG

WR373:  JMP     H37DONE         ;EXIT THRU H37DONE

        PAGE
;
; H37E -- CHECK IF I SHOULD DO RETRIES.
;
;    ENTRY:  (ERRCNT) = RETRY COUNTER
;    EXIT:   PSW/C = 0 IF DO RETRY
```

```
;                    = 1 IF DON'T
;              (ERRCNT) UPDATED
;    USES:   ALL
;
;  I/O RECOVERY PROCEDURE:
;    1) RETRY I/O OPERATION
;    2) RESTORE HEAD, THEN RETRY I/O OPERATION
;    3) RETRY I/O OPERATION
;    4) FLAG HARD ERROR
;

H37E:
        STA     ERRTYP          ;SAVE ERROR CODE

        ANI     FDSNRD+FDSWPV   ;CHECK FOR NOT READY OR WRITE PROTECT
        JNZ     H37E1           ; BR IF YES - NO RETRIES

        LDA     SET37A          ;IF NOT DOING SELECT
        ANA     A
        JNZ     H37E0
        LHLD    SECNT37         ;BUMP SOFT ERROR COUNTER
        INX     H
        SHLD    SECNT37

H37E0:  LXI     H,ERRCNT        ;UPDATE RETRY COUNTER
        DCR     M
        JZ      H37E1           ; BR IF EXHAUSTED RETRIES

        MOV     A,M
        CPI     2
        CZ      RST37           ;RESTORE HEAD BEFORE TRYING AGAIN

        XRA     A               ;INDICATE TRY AGAIN
        RET

H37E1:  LXI     H,H37MSG
        CALL    PRTERR          ;PRINT EXTENDED ERROR MSG

        STC                     ;INDICATE DON'T TRY AGAIN
        RET

H37MSG  DB      'H37',0

        PAGE
;
;  H37 INTERRUPT HANDLER.
;
;    USES INTERRUPT LEVEL 4
;    THE INTERRUPT HANDLER RETURNS CONTROL TO THE USER PROGRAM
;     AT THE ADDRESS IN 'H37IRET'.
;
;    EXIT:  (A) = STATUS BYTE
;    USES:  A,F
;

H37ISR: MVI     A,10            ;DELAY AWHILE TO LET STATUS SETTLE
```

```
H37ISR1: DCR     A
         JNZ     H37ISR1

         IN      FD$STA          ;INPUT STATUS TO CLEAR INTERRUPT
         XTHL                    ;SAVE (HL)  DISCARD RET ADDR
         LHLD    H37IRET         ;GET NEW RETURN ADDRESS
         XTHL                    ;RESTORE (HL)  SET NEW RET ADDR
         EI                      ;RE-ENABLE INTERRUPTS
         RET

         PAGE
;
;  SDP37 -- SET DEVICE PARAMETERS
;
;    ENTRY:  (HSTDPB) = ADDR OF HEATH EXTENSIONS FOR DRIVE
;            (HSTTRK) = TRACK
;            (HSTSEC) = SECTOR (0 TO SPT-1)
;    USES:   ALL
;
SDP37:
         CALL    ONH37           ;SELECT DRIVE

         LDA     SET37A
         ANA     A
         MVI     A,4             ;SET TRY COUNTER
         JZ      SDP371
         RAR                     ;ONLY 2 TRIES IF DOING SELECT
SDP371:  STA     ERRCNT

         LDA     HSTTRK          ;GET TRACK #
         STA     TRACK
         MVI     B,0             ;DEFAULT TO SIDE 0
         LHLD    HSTDPB
         MOV     A,M
         ANI     DPE2S           ;CHECK IF MEDIA IS DOUBLE SIDED
         JZ      SDP374          ;BR IF NOT
         LXI     H,TRACK         ;IS
         MOV     A,M             ; DOUBLE SIDED
         ANA     A               ;  THEREFORE DIVIDE TRACK #
         RAR                     ;  BY 2 TO GET REAL TRACK #
         MOV     M,A
         JNC     SDP374          ;  EVEN TRACKS ON SIDE 0
         MVI     B,FDFSS1        ;  ODD TRACKS ON SIDE 1
SDP374:  MOV     A,B
         STA     SIDE            ;SAVE SIDE
         LDA     HSTSEC          ;GET SECTOR #
         INR     A               ;MAKE 1 - SPT
         STA     SECTOR

         LHLD    HSTDPB          ;SET TRKPT FOR THIS UNIT
         LXI     D,DPETRK-DPEHTH
         DAD     D
         SHLD    TRKPT
         MOV     A,M             ;GET CURRENT TRACK
         RAL                     ;IF MSB==1 THEN DON'T KNOW WHERE
```

```
        JC      RST37           ;  HEAD IS, SO RESTORE
        MVI     A,FD$TS
        OUT     FD$INT          ;ACCESS T/S REGS
        MOV     A,M             ;UPDATE TRACK REG
        OUT     FD$TRK          ;  WITH CURRENT VALUE FOR THIS DRIVE
        RET

        PAGE
;
;  ONH37 -- TURN ON MOTOR, SELECT DRIVE, AND SET SETTLE DELAY COUNTERS
;

ONH37:
        LXI     H,0
        SHLD    DLYMO37
        LHLD    HSTDPB          ;GET THE DRIVE SELECT CODE AND DENSITY
        MOV     A,M
        ANI     DPEDD           ;CHECK BIT DENSITY
        JZ      ONH37A          ; BR IF SINGLE
        MVI     A,CONMFM        ;SET DOUBLE BIT DENSITY CONTROL FLAG
ONH37A: INX     H
        ORA     M               ;OR IN UNIT SELECT
        ORI     CONMO+CONIRQ    ;OR THE MOTOR & IRQ ON
        OUT     FD$CON
        MOV     B,A             ;SAVE THIS VALUE TEMPORARILY
        LXI     H,H37CTL        ;GET THE CURRENT VALUE OF THE CONTROL PORT
        MOV     A,M
        ANI     CONMO           ;IF THE MOTOR WAS ON
        JNZ     ONH37B          ;  THEN WE DON'T HAVE TO WAIT FOR IT TO COME UP
        PUSH    H
        LHLD    HSTDPB          ;FIND OUT HOW FAST THE DRIVE COMES UP TO SPEED
        LXI     D,DPESEK-DPEHTH
        DAD     D
        MOV     A,M
        POP     H
        RAL                     ;SETS CARRY IF UP FAST
        IF      DPEMO-10000000B
%:      DPEMO NE 10000000B
        ENDIF
        MVI     A,(1000+3)/4+1  ;NORMAL TIMING (APPROX 1 SECOND)
        JNC     ONH37C
        RAR                     ;NEW DRIVES UP IN 1/4 THE TIME
        RAR
        ANI     03FH
        JMP     ONH37C
ONH37B  MOV     A,M             ;GET THE OLD VALUE OF THE CONTROL PORT
        ANI     CONDS0+CONDS1+CONDS2+CONDS3     ;CHECK SELECT DRIVE(S)
        ANA     B               ;CHECK TO SEE IF SAME HEAD ALREADY DOWN
        MVI     A,0
        JNZ     ONH37C          ;YES, ALREADY LOADED, NO DELAY
        MVI     A,(50+3)/4+1    ;MUST DELAY FOR HEAD LOAD
ONH37C: STA     DLYW
        MOV     M,B             ;SET NEW VALUE OF CONTROL PORT

        RET
```

```
                PAGE
                ;
                ;  RST37 -- RESTORE HEAD
                ;
                ;    ENTRY:  (TRKPT) = ADDRESS OF DPETRK SLOT IN HEATH EXTENSION OF DPE
                ;    EXIT:   VIA H37WAIT
                ;    USES:   ALL
                ;

RST37:          CALL    WBS37
                LHLD    TRKPT           ;INDICATE TRACK IS 0
                MVI     M,0

                INX     H               ;HL -> STEP RATE
                IF      (DPETRK+1)-DPESEK
%:              DPESEK NE (DPETRK+1)
                ENDIF
                MOV     A,M
                ANI     OFFH-DPEMO
                ORI     FDCRST
                JMP     H37WAIT         ;ISSUE COMMAND

                PAGE
                ;
                ;  MOUNT H37 MEDIA.
                ;

MNTH37:         LHLD    HSTDPB          ;SET TRACK POINT TO UNKNOWN
                LXI     D,DPETRK-DPEHTH
                DAD     D
                MVI     M,DPEUNK

;+              JMP     RESH37          ;RESET DRIVE


                ;
                ;  RESET H37
                ;

RESH37:         MVI     A,FD$CD         ;ACCESS C/D REGS
                OUT     FD$INT
                MVI     A,FDCFI+FDFINI  ;TERMINATE ANY PRESENT ACTIVITY
                OUT     FD$CMD
                XRA     A
                OUT     FD$CON          ;TURN OFF HARDWARE
                STA     H37CTL
                LXI     H,0
                SHLD    DLYMO37
                SHLD    H37IRET
                IN      FD$DAT          ;CLEAR ANY PENDING DRQ

                MVI     A,10            ;DELAY FOR AWHILE
RESH371:                                ;  TO LET STATUS SETTLE
                DCR     A
```

```
        JNZ     RESH371
        IN      FD$STA          ;CLEAR ANY PENDING IRQ

        RET

        PAGE
;
;  SDT37 -- SEEK DESIRED TRACK AND SET SECTOR REG FOR DESIRED SECTOR
;
;    ENTRY:  (TRACK) = TRACK
;            (SECTOR) = SECTOR
;    USES:   ALL
;

SDT37:  LHLD    TRKPT
        MOV     A,M             ;GET CURRENT TRACK
        LXI     H,TRACK         ;GET DESIRED TRACK #
        CMP     M
        JZ      SDT372          ;ALREADY AT DESIRED TRACK

        CALL    WBS37
        MOV     A,M
        CALL    SDT376          ;DO SEEK

        LHLD    HSTDPB          ;CHECK IF 48TPI R/O MEDIA IN 96TPI DRIVE
        MOV     A,M
        ANI     DPE48RO
        JZ      SDT371          ; BR IF NOT -- NO NEED TO DOUBLE STEP

        MVI     A,FD$TS         ;GO THROUGH SEEK PROCEDURE AGAIN A
        OUT     FD$INT          ; A SECOND TIME
        LHLD    TRKPT
        MOV     A,M
        OUT     FD$TRK          ;RESET CONTROLLER'S TRACK REG
        LDA     TRACK
        CALL    SDT376          ;DO 2ND SEEK

SDT371: MVI     A,FD$TS         ;ACCESS T/S REGS
        OUT     FD$INT
        LDA     TRACK           ;OUTPUT DESIRED TRACK TO TRACK REG
        OUT     FD$TRK          ; THIS IS DONE IN CASE SEEK ABORTED
                                ; WHEN IT TRIED TO STEP TO NEGATIVE
                                ; TRACK LEAVING TRACK REGISTER
                                ; CONTAINING 0.
        LHLD    TRKPT
        MOV     M,A             ;UPDATE IN MEMORY TRACK VALUE

        MVI     A,(15+3)/4+1    ;SETTLE DELAY COUNT (15 MS)
        LXI     H,DLYW          ;CHECK IF DELAY COUNTER
        CMP     M               ;  IS >= STEP SETTLE TIME
        JC      SDT372          ;IT IS
        MOV     M,A             ;FORCE DELAY FOR SETTLE TIME

SDT372: MVI     A,FD$TS
        OUT     FD$INT          ;ACCESS T/S REGS
        LDA     SECTOR
```

```
                OUT     FD$SEC

        SDT373: LDA     DLYW            ;WAIT UNTIL MOTOR IS UP TO SPEED,
                ORA     A               ; AND HEAD SETTLES DUE TO
                JNZ     SDT373          ;  LOADING OR STEPPING

                RET

        ;  DO SEEK
        ;    (A) = DESIRED TRACK

        SDT376: MOV     B,A
                MVI     A,FD$CD
                OUT     FD$INT          ;ACCESS C/D REGS
                MOV     A,B
                OUT     FD$DAT          ;TELL CONTROLLER DESIRED TRACK

                LHLD    TRKPT           ;GET STEP RATE
                INX     H
                MOV     A,M
                ANI     OFFH-DPEMO
                ORI     FDCSEK+FDFHLB   ; OR IN COMMAND
                JMP     H37WAIT         ;ISSUE COMMAND AND WAIT
                                        ; (RET THRU H37WAIT)

                PAGE
        ;
        ;  RDYH37 -- CHECK IF UNIT READY
        ;            1. DRIVE IS AVAILABLE
        ;            2. CORRECT TYPE OF MEDIA HAS BEEN INSERTED
        ;
        ;  ENTRY:  NONE
        ;  EXIT:   PSW/C = 0 IF READY
        ;                = 1 IF NOT READY
        ;  USES:   ALL
        ;
        ;  THE NUMBER OF HOLE TRANSITIONS IS COUNTED IN THE TIME IT TAKES
        ;  FOR 2 REVOLUTIONS (400MS).  IF THE DISK INSERTED IS SOFT SECTORED,
        ;  THEN 2 HOLES PLUS/MINUS 1 (4+-2 TRANSITIONS) SHOULD BE SEEN.
        ;
        RDYH37:
                CALL    ONH37           ;TURN OF MOTOR AND SELECT DRIVE

                LXI     H,DLYW          ;WAIT UNTIL DRIVE IS UP TO SPEED
        RDYH37A:
                MOV     A,M
                ANA     A
                JNZ     RDYH37A

                MVI     A,FD$CD         ;ACCESS C/D REGS
                OUT     FD$INT
                MVI     A,FDCFI+FDFINI  ;FORCE TYPE STATUS
                OUT     FD$CMD
                MVI     A,10
        RDYH37B:
```

```
        DCR     A               ;DELAY AWHILE TO LET CONTROLLER SETTLE
        JNZ     RDYH37B

        LXI     H,TICCNT        ;GET TIME VALUE
        MVI     A,200
        ADD     M
        MOV     B,A             ;(B) = TIME VALUE

        MVI     C,0             ;(C) = HOLE COUNTER
        MOV     D,C             ;(D) = INIT HOLE STATUS TO NO HOLE

RDYH37C:
        IN      FD$STA          ;GET HOLE STATUS
        ANI     FDSIND
        CMP     D               ;CHECK IF CHANGE IN STATUS
        JZ      RDYH37D         ; BR IF NO CHANGE

        MOV     D,A             ;SAVE NEW STATUS
        INR     C               ;COUNT TRANSITION

RDYH37D:
        MOV     A,B             ;CHECK IF TIME UP
        CMP     M
        JNZ     RDYH37C         ; BR IF NOT
        MOV     A,C             ;TIME UP -- CHECK # OF HOLES
        CPI     1*2
        JC      RDYH37E         ;IF < 1 THEN ERROR
        CPI     3*2+1
        CMC
        JNC     RDYH37F         ;IF <= 3 THEN OK

RDYH37E:
        MVI     A,FDSNRD        ;ERROR CODE FOR UNIT NOT READY
        CALL    H37E            ;REPORT ERROR

RDYH37F:
        JMP     H37DONE         ;RETRUN VIA 'H37DONE'


;
;  WBS37 -- WAIT BEFORE STEPPING
;           IT IS A DRIVE REQUIREMENT THAT AFTER A WRITE OPERATION, A STEP
;           COMMAND SHOULD NOT BE EXECUTED BEFORE 1 MS AFTER THE WRITE.
;           THEREFORE, TO INSURE THIS, WBS37 IS USED BEFORE ANY STEP
;           OPERATIONS TO DELAY FOR APPROXIMATELY 1 MS.
;
;    USES:  A,F
;

WBS37:  MVI     A,150
WBS371: DCR     A
        JNZ     WBS371
        RET
```

```
                ENDIF
                PAGE
```

```
                IF      H47T

        ;       H47 DISK DEVICE DRIVER                  23 JUL 81

H47DVD: JMP     SETD47          ;SELECT DISK ENTRY POINT
        JMP     RD47            ;READ
        JMP     WR47            ;WRITE
        JMP     RESH47          ;RESET
        XRA A ! RET ! NOP       ;MOUNT

        ;
        ;       8 INCH DISK DESCRIPTORS
        ;
        ;       DW      &XLATE$TABLE    ;POINTER TO TRANSLATE TABLE
        ;       DW      &PARAM$BLOCK    ;POINTER TO PARAMETER BLOCK
        ;       DB      DENSITY AND SIDES
        ;       DB      RECORDS$PER$SECTOR
        ;       DB      RECORDS$PER$ALLOCATION

H47PMS: DW      XLTOS,DPB0SS    ;FOR    SINGLE DENSITY  SINGLE SIDED
        DB      0,1,8

        DW      XLTOS,DPB0SD    ;       SINGLE          DOUBLE
        DB      DPE2S,1,16

        DW      XLTOD,DPB0DS    ;       DOUBLE          SINGLE
        DB      DPEDD,2,16

        DW      XLTOD,DPB0DD    ;       DOUBLE          DOUBLE
        DB      DPEDD+DPE2S,2,16

        IF      H47ED

        DW      0,DPB0ES        ;       EXTENDED        SINGLE
        DB      DPEED+DPEDD,8,16

        DW      0,DPB0ED        ;       EXTENDED        DOUBLE
        DB      DPEED+DPEDD+DPE2S,8,16

        ENDIF
        PAGE
SETD47: MVI     A,DRAS
        CALL    WCD

        LHLD    DPBX
        INX     H
        MOV     A,M
        ORI     001H
        CALL    WBD

        CALL    W4TR
        CALL    H47IND          ;READ THE AUXILIARY STATUS INFORMATION
        PUSH    PSW
```

```
        ANI     03H             ;FIND SECTOR LENGTH
        CPI     2
        IF      H47ED
        JC      SETD1           ;IF LEN <> 128 OR 256
        MVI     A,2             ;  THEN MAKE INDEX 2
        ELSE
        JNC     SETD9           ;IF LEN <> 128 OR 256 THEN ERROR
        ENDIF

SETD1:  ADD     A               ;*2 TO ALLOW FOR SINGLE/DOUBLE DENSITY
        MOV     D,A             ;SAVE INDEX TO DATE

        POP     PSW
        ANI     10H             ;CHECK FOR SIDE 1
        MOV     A,D
        JZ      SETD2

        ORI     1               ;FLAG SIDE 1 AVAILABLE

SETD2:  MOV     B,A
        ADD     A               ;*2
        ADD     B               ;*3
        ADD     A               ;*6
        ADD     B               ;*7

        LXI     H,H47PMS
        CALL    DADA

        XCHG                    ;DE NOW POINTS TO PARAMETERS FOR THIS DISK
        LHLD    SETDSKC         ;HL POINTS TO DPE FOR THIS DRIVE
        LDAX    D               ;STORE THE NEW TRANSLATE TABLE
        MOV     M,A
        INX     D
        INX     H
        LDAX    D
        MOV     M,A
        INX     D

        PUSH    D
        LXI     D,DPEDPB-1      ;BUMP HL TO POINT TO DPB POINTER
        DAD     D
        POP     D

        LDAX    D               ;NOW SET UP NEW DPB POINTER
        MOV     M,A
        INX     D
        INX     H
        LDAX    D
        MOV     M,A
        INX     D

        LHLD    DPBX            ;GET POINTER TO THIS DRIVE
        MOV     A,M             ;FLAG DENSITY AND SIDEDNESS
        ANI     OFFH-DPEED-DPEDD-DPE2S
        MOV     B,A
        LDAX    D
```

```
        ORA     B
        MOV     M,A
        INX     D               ;RECORDS PER SECTOR
        INX     H
        INX     H
        LDAX    D
        MOV     M,A
        INX     D               ;AND RECORDS PER ALLOCATION
        INX     H
        LDAX    D
        MOV     M,A

        XRA     A               ;INDICATE NO ERRORS
        RET

        IF      NOT H47ED
SETD9:  STC                     ;INDICATE ERROR
        RET
        ENDIF

        PAGE
;
;  RESET H47.
;

RESH47: MVI     A,DCRES
        CALL    H47OUTC
        CALL    W4DONE          ;WAIT FOR THE CONTROLLER TO BE DONE
        XRA     A               ;CLEAR CONTROL REG
        CALL    H47OUTC
        RET

        PAGE
RD47:   MVI     A,DRD           ;DO A BUFFERED READ
        CALL    SET47
        JC      RDERR

RDH3:   CALL    H47INS          ;GET THE CONTROL PORT
        ANI     DSTR+DSDONE+DSERR ;WAIT FOR ANY LINE OF INTEREST
        JZ      RDH3            ;NOTHING YET
        ANI     DSDONE+DSERR    ;LOOK FOR ERROR OR END OF SECTOR
        JNZ     RDH4
        CALL    H47IND          ;GET DATA BYTE
        MOV     M,A             ;SAVE IT
        INX     H               ;BUMP MEMORY POINTER
        JMP     RDH3

RDH4:   CALL    H47INS          ;REREAD IN CASE ERROR WAS LATE
        ANI     DSERR           ;IF THERE WAS NO ERROR
        RZ                      ;  THEN RETURN

RDERR:
ER47:   MVI     A,DRS           ;READ DISK SUBSYSTEM STATUS
        CALL    WCD
        CALL    W4TR
        CALL    H47IND
```

```
        STA     ERRTYP          ;SAVE ERROR CODE

        LXI     H,H47MSG        ;PRINT EXTENDED ERROR MSG
        CALL    PRTERR

        CALL    RESH47          ;RESET THE DISK SUBSYSTEM

        MVI     A,0FFH          ;FLAG ERROR
        STA     ERFLAG
        RET

H47MSG  DB      'H47',0

        PAGE
WR47:   MVI     A,DWR           ;DO A H47 BUFFERED WRITE
        CALL    SET47           ;DO COMMON SETUP
        JC      WRERR
        CALL    H47INS          ;CHECK ON DISK STATUS
        ANI     DSERR
        JNZ     WRERR           ;WHOOPS
WRH3:   CALL    H47INS          ;CHECK DISK STATUS
        ANI     DSTR+DSDONE+DSERR
        JZ      WRH3
        ANI     DSDONE+DSERR
        JNZ     WRH4
        MOV     A,M             ;WRITE OUT THE NEXT BYTE
        CALL    H47OUTD
        INX     H               ;BUMP MEMORY POINTER
        JMP     WRH3

WRH4:   CALL    H47INS          ;REREAD IN CASE ERROR WAS LATE
        ANI     DSERR
        RZ

WRERR:  JMP     ER47            ;PRINT THE ERROR MESSAGE AND CODE

        PAGE
;
;       SET47 - COMMON H47 SETUP FOR READ/WRITE
;

SET47:  CALL    WCD             ;WRITE THE COMMAND IN A TO DISK CONTROLLER
        RC

        LHLD    HSTDPB
        MOV     A,M
        RAR                     ;IF LEAST SIGNIFICANT BIT IS 1 THEN DS
        IF      DPE2S-1
%:      DPE2S NE 1
        ENDIF
        MVI     A,0
        STA     SIDE
        LDA     HSTTRK
        JNC     SET472          ;DISK IS SINGLE SIDED
        RRC                     ;DIVIDE TRACK BY TWO, SIDE INTO MSB
        PUSH    PSW
```

```
                ANI     080H
                STA     SIDE
                POP     PSW
                ANI     07FH

        SET472: CALL    WBD
                RC

                LHLD    HSTDPB
                INX     H               ;HL POINTS TO SPECIFIED UNIT CODE

                LDA     HSTSEC          ;GET DESIRED SECTOR
                INR     A               ;IBM DISK SECTORS NUMBERED 1 TO N
                ORA     M               ;OR IN THE SELECTED DRIVE

                LXI     H,SIDE
                ORA     M

                CALL    WBD
                RC

                LXI     H,HSTBUF        ;(TEMPORARY) DESTINATION FOR DATA
                RET

                PAGE
        ;       WCD - WRITE COMMAND TO DISK
        ;               DISK SHOULD BE "DONE" TO ACCEPT A NEW COMMAND

        WCD:    CALL    W4DONE          ;WAIT UNITL DONE BEFORE COMMANDING
                RC                      ;ERROR - DONE TIMEOUT
                CALL    H47OUTD         ;SEND COMMAND
                CALL    W4ND
                RET

        ;
        ;       WBD - WRITE BYTE TO DISK
        ;               A BYTE CAN BE SENT WHEN TR IS ASSERTED

        WBD:    CALL    W4TR
                RC
                CALL    H47OUTD
                RET

        ;
        ;       W4DONE - WAIT FOR DONE TO BE ASSERTED
        ;               TIME OUT IN ABOUT 4 SEC, RETURN WITH C SET

        W4DONE: PUSH    PSW
                PUSH    B
                LXI     B,0FFFFH
        W4D1:   CALL    H47INS
                ANI     DSDONE          ;IS IT DONE YET?
                JNZ     W4D2            ;YES, CLEAN UP AND RETURN
                DCX     B               ;DECREMENT TIME OUT TIMER
                MOV     A,B             ;IS IT ZERO YET?
                ORA     C
```

```
                JNZ     W4D1            ;NO, WAIT A WHILE LONGER
                POP     B               ;TIME OUT - RETURN WITH C SET
                POP     PSW
                STC
                RET

        W4D2:   POP     B
                POP     PSW
                ORA     A               ;CLEAR CARRY
                RET

        ;
        ;       W4ND - WAIT FOR NOT DONE
        ;

        W4ND:   PUSH    PSW
        W4ND1:  CALL    H47INS
                ANI     DSDONE
                JNZ     W4ND1
                POP     PSW             ;RETURN AFTER DONE REMOVED
                RET

        ;
        ;       W4TR - WAIT FOR TR TO BE ASSERTED
        ;

        W4TR:   PUSH    PSW
        W4TR1:  CALL    H47INS          ;GET THE DISK STATUS
                ANI     DSDONE+DSTR
                JZ      W4TR1
                ANI     DSDONE
                JNZ     W4TR2
                POP     PSW
                ORA     A
                RET
        W4TR2:  POP     PSW
                STC
                RET

                PAGE
        H47INS: IN      78H             ;INPUT STATUS BYTE
        H47INS1 EQU     $-1
                RET

        H47OUTC: OUT    78H             ;OUTPUT CONTROL BYTE
        H47OUTC1 EQU    $-1
                RET

        H47IND: IN      79H             ;INPUT DATA BYTE
        H47IND1 EQU     $-1
                RET

        H47OUTD: OUT    79H             ;OUTPUT DATA BYTE
        H47OUTD1 EQU    $-1
                RET
```

```
ENDIF
PAGE
```

```
                IF      H67T

;               H67 DEVICE DRIVER MODULE                        04 SEP 81
;
;*              JWTITTSLER / BILL ZURNEY
;               HEATH/ZENITH SOFTWARE GROUP
;               HILLTOP ROAD
;               SAINT JOSEPH, MICHIGAN  49085
;

H67DVD: JMP     SET67           ;SELECT DISK ENTRY POINT
        JMP     RD67            ;READ
        JMP     WR67            ;WRITE
        JMP     RESH67          ;RESET
        XRA A ! RET ! NOP       ;MOUNT

SET67:
        XRA     A               ;1ST SECTOR WILL BE READ
        STA     HSTSEC          ; (SECTOR # 0 TO SPT-1)

        LHLD    DPBX
        MOV     A,M
        ANI     DPETYPF
        CPI     DPEH67F         ;CHECK FOR FLOPPY
        JZ      SET671          ;BR IF YES

;  1ST TIME SELECTION FOR HARD DISK.

        LXI     H,0             ;READ LABEL
        SHLD    HSTTRK
        CALL    RD67
        JNZ     SET679          ; BR IF ERROR

        CALL    CHKLAB          ;CHECK CHECKSUM OF LABEL
        JNZ     SET679          ; BR IF INCORRECT CHECKSUM

        IF      PARTITN

        LHLD    DPBX            ;CHECK IF LABEL'S BEGINNING OF
        LXI     B,DPETRK-DPEHTH ; PARTITION SECTOR NUMBER MATCHES
        DAD     B               ;  DRIVE TABLE'S
        CALL    HLIHL
        XCHG
        LHLD    HSTBUF+LABHTH+DPETRK-DPEHTH
        CALL    CPHLDE
        JNZ     SET679          ;BR IF NOT -- PARTITION HAS BEEN MOVED

        LHLD    DPBX            ;CHECK IF LABEL'S LAST SECTOR # + 1
        LXI     B,DPEUPB-DPEHTH ; OF PARTITION MATCHES
        DAD     B               ;  DRIVE TABLE'S
        CALL    HLIHL
        XCHG
        LHLD    HSTBUF+LABHTH+DPEUPB-DPEHTH
        CALL    CPHLDE
        JNZ     SET679          ;BR IF NOT -- PARTITION SIZE HAS CHANGED
```

```
                ENDIF

                LHLD    DPBX            ;UPDATE CP/M RECORDS PER ALLOCATION BLOCK
                LXI     B,DPERPAB-DPEHTH
                DAD     B
                LXI     D,HSTBUF+LABHTH+DPERPAB-DPEHTH
                LDAX    D
                MOV     M,A

                LHLD    SETDSKC         ;UPDATE DISK PARAMETER BLOCK VALUES
                LXI     B,DPEDPB
                DAD     B
                CALL    HLIHL
                LXI     D,HSTBUF+LABDPB
                MVI     C,DPBL
                CALL    MOVEITX

                JMP     SET674

        ;  1ST TIME SELECTION FOR FLOPPY.

        SET671:
                LHLD    DPBX            ;GET ADDR OF HEATH EXTENSIONS FOR DRIVE
                MOV     A,M             ;GET DENSITY FROM TABLE
                ANI     DPEDD
                JZ      SET671A         ; CURRENT DENSITY IS SINGLE
                MVI     A,HDFDEN        ;DOUBLE DENSITY

        SET671A:
                STA     SET67A          ;ASSUME CURRENT DENSITY
                CALL    FDD67           ;SET CURRENT DENSITY/SINGLE SIDED
                JNZ     SET679          ;BR IF ERROR
                LXI     H,1             ;TRY READING TRACK 1 SECTOR 1
                SHLD    HSTTRK
                CALL    RD67
                JZ      SET672          ;BR IF NOT READ ERROR

                LDA     ERRTYP
                CPI     HDEIAM          ;CHECK FOR ID ADDR MARK NOT FOUND
                JNZ     SET679          ;BR IF NOT
                LXI     H,SET67A        ;TRY OTHER DENSITY
                MOV     A,M
                XRI     HDFDEN
                MOV     M,A
                CALL    FDD67           ;SET OTHER DENSITY/SINGLE SIDED
                CALL    RD67            ;VERIFY OTHER DENSITY
                JNZ     SET679          ; BR IF NOT

        SET672: LXI     H,SET67A
                MOV     A,M
                ORI     HDFSID          ;TRY DOUBLE SIDED
                MOV     M,A
                CALL    FDD67           ;SET DENSITY/SIDE DESCRIPTION
                JNZ     SET679          ;BR IF ERROR
```

```
        CALL    RD67            ;TRY READING 2ND SIDE
        JZ      SET673          ;BR IF NO READ ERROR
        LDA     ERRTYP
        CPI     HDEDNR          ;CHECK FOR DRIVE NOT READY
        JNZ     SET679          ;BR IF NOT
        LXI     H,SET67A
        MOV     A,M
        ANI     OFFH-HDFSID     ;BACK OFF TO SINGLE SIDED
        MOV     M,A

SET673: LDA     SET67A
        MOV     B,A
        ADD     A               ;*2
        ADD     B               ;*3
        ADD     A               ;*6
        ADD     B               ;*7
        LXI     H,H67PMS
        CALL    DADA            ;HL = ADDR OF SPECIFIC DISK TYPE VALUES
        XCHG                    ;DE NO POINTS TO PARAMETERS FOR THIS DISK
        LHLD    SETDSKC         ;HL POINTS TO DPE FOR THIS DRIVE

        LDAX    D               ;UPDATE XLATE TABLE ADDR
        MOV     M,A
        INX     D
        INX     H
        LDAX    D
        MOV     M,A
        INX     D
        LXI     B,DPEDPB-1      ;BUMP HL TO POINT TO DPB POINTER
        DAD     B
        LDAX    D               ;NOW SET UP NEW DPB POINTER
        MOV     M,A
        INX     D
        INX     H
        LDAX    D
        MOV     M,A
        LHLD    DPBX            ;GET POINTER TO DRIVE SPECIFIC VALUES
        XCHG                    ;(DE)=DRIVE SPECIFIC VALUES POINTER
                                ;(HL)=H67 DISK DESCRIPTORS POINTER
        INX     H
        LDAX    D
        ANI     OFFH-DPEDD-DPE2S
        ORA     M
        STAX    D               ;UPDATE DENSITY/SIDE FLAGS
        INX     D
        INX     D
        INX     H
        MOV     A,M
        STAX    D               ;RECORDS PER SECTOR
        INX     H
        INX     D
        MOV     A,M
        STAX    D               ;RECORDS PER ALLOCATION

        LDA     SET67A
```

```
        CALL    FDD67           ;SET CORRECT DENSITY/SIDE DESCRIPTION
        JNZ     SET679          ;BR IF ERROR

SET674:
        XRA     A               ;INDICATE NO ERRORS
        RET

SET679: STC                     ;INDICATE ERROR
        RET

;  DISK DESCRIPTORS
;
;       DW      &XLATE$TABLE    ;POINTER TO TRANSLATE TABLE
;       DW      &PARAM$BLOCK    ;POINTER TO PARAMETER BLOCK
;       DB      DENSITY AND SIDES
;       DB      RECORDS$PER$SECTOR
;       DB      RECORDS$PER$ALLOCATION

H67PMS: DW      XLTOS,DPB0SS    ;SINGLE DENSITY/SINGLE SIDED
        DB      0,1,8

        DW      XLTOS,DPB0SD    ;SINGLE DENSITY/DOUBLE SIDED
        DB      DPE2S,1,16

        DW      XLTOD,DPB0DS    ;DOUBLE DENSITY/SINGLE SIDED
        DB      DPEDD,2,16

        DW      XLTOD,DPB0DD    ;DOUBLE DENSITY/DOUBLE SIDED
        DB      DPEDD+DPE2S,2,16

SET67A: DS      1               ;DENSITY/SIDE DESCRIPTION VALUE

        PAGE
;
;  RESET H67 CONTROLLER.
;

RESH67: MVI     A,HDFRES
        CALL    H67OUTC
        XRA     A
        CALL    H67OUTC
        RET

        PAGE
;
;  RECALIBRATE HEAD.
;

RCL67:  MVI     A,HDCRCL
        STA     CMDBUF+HDOOP
        CALL    SETUP3
        MVI     A,1             ;INDICATE DON'T DO RS ON ERROR
        STA     RS67B
        JMP     CMPSTAT

        PAGE
```

```
                ;
                ; READ SENSE BYTES.
                ;
                ;   ENTRY:  COMMAND BUFFER ALREADY CONTAINS UNIT SELECT VALUE
                ;   EXIT:   4 SENSE BYTES ARE AT 'RS67A'
                ;           'ERRTYP' CONTAINS ERROR TYPE & CODE
                ;           (A) CONTAINS ERROR TYPE & CODE
                ;   USES:   ALL
                ;

RS67:   MVI     A,HDCRS
        STA     CMDBUF+HDOOP
        XRA     A
        STA     CMDBUF+HDOCON
        CALL    SETUP3
        MVI     A,1             ;INDICATE DON'T DO RS IF ERROR
        STA     RS67B
        LXI     H,RS67A
        MVI     M,0

        IF      H67BLKIO
        MVI     A,4             ;NUMBER OF BYTES TO READ
        STA     RD673           ; MODIFY COUNT INSTRUCTION
        ENDIF
        CALL    RD671

        MVI     A,0             ;ASSUME NO STATUS IN CASE OF ERROR ON RS
        JNZ     RS671
        LDA     RS67A
RS671:  ANI     HDSET+HDSEC
        STA     ERRTYP
        RET

RS67A   DS      4
RS67B   DS      1

        PAGE
                ;
                ; FLOPPY DISK DESCRIPTION.
                ;
                ;   ENTRY:  (A) = TRACK FORMAT CODE
                ;   EXIT:   PSW/Z = 0 IF ERROR
                ;                = 1 IF NO ERROR
                ;           EXIT VIA 'CMPSTAT'
                ;   USES:   ALL
                ;

FDD67:  STA     CMDBUF+HD6TFC
        LHLD    HSTDPB
        INX     H
        MOV     A,M
        STA     CMDBUF+HD6LUN
        MVI     A,HDCFDD
        STA     CMDBUF+HD6OP
        CALL    SETUP3
        JMP     CMPSTAT
```

```
                PAGE
                ;
                ; READ SECTOR.
                ;
                ;   ENTRY:  (HSTDPB) = ADDR OF DPE HEATH EXTENSIONS FOR DRIVE
                ;           (HSTTRK) = TRACK
                ;           (HSTSEC) = SECTOR (0 TO SPT-1)
                ;   EXIT:   PSW/Z = 0 IF ERROR
                ;                 = 1 IF NO ERROR
                ;           'ERRFLAG' = 0 IF NO ERROR
                ;                     <> 0 IF ERROR
                ;           'HSTBUF' CONTAINS SECTOR
                ;   USES:   ALL
                ;
RD67:           MVI     A,HDCRD         ;READ COMMAND
                CALL    SETUP           ;SETUP COMMAND BUFFER, GET ATTENTION, SEND CMD
                RNZ                     ; RET IF ERROR
                LXI     H,HSTBUF        ;POINTER TO DATA STORAGE
                IF      H67BLKIO
                MVI     A,128           ;BLOCK COUNT
                STA     RD673           ; MODIFY COUNT INSTRUCTION
                ENDIF

                ; LOW LEVEL I/O READ ROUTINE
                ;
                ;   ENTRY:  (HL)=BUFFER ADDRESS
                ;   EXIT:   VIA CMPSTAT
                ;   USES:   ALL

                IF      H67BLKIO

RD671:          LDA     H67IND1         ;GET DATA INPUT PORT #
                MOV     C,A
RD672:          CALL    H67INS          ;GET THE BUS STATUS
                ANA     A               ;LOOK FOR REQ
                IF      HDBREQ-10000000B
%:              HDBREQ NE 10000000B
                ENDIF
                JP      RD672
                ANI     HDBCMD          ;CHECK FOR COMPLETION
                JNZ     CMPSTAT
                MVI     B,0             ;BLOCK COUNT
RD673           EQU     $-1             ; (MODIFIED BY CALLING ROUTINE)
                DB      0EDH,0B2H       ;DO BLOCK INPUT
                JMP     RD672

                ELSE

RD671:          CALL    H67INS          ;GET THE BUS STATUS
                ANA     A               ;LOOK FOR REQ
                IF      HDBREQ-10000000B
%:              HDBREQ NE 10000000B
                ENDIF
                JP      RD671
```

```
        ANI     HDBCMD          ;CHECK FOR COMPLETION
        JNZ     CMPSTAT
        CALL    H67IND          ;INPUT DATA FROM THE CONTROLLER
        MOV     M,A             ;STORE IT IN THE HOST BUFFER
        INX     H
        JMP     RD671

        ENDIF

        PAGE
;
;  WRITE SECTOR.
;
;     ENTRY:  (HSTDPB) = ADDR OF DPE HEATH EXTENSIONS FOR DRIVE
;             (HSTTRK) = TRACK
;             (HSTSEC) = SECTOR (0 TO SPT-1)
;     EXIT:   PSW/Z = 0 IF ERROR
;                     1 IF NO ERROR
;             'ERFLAG' = 0 IF NO ERROR
;                      <> 0 IF ERROR
;             DATA AT 'HSTBUF' WRITTEN
;     USES:   ALL
;
WR67:   MVI     A,HDCWR         ;WRITE COMMAND
        CALL    SETUP           ;SETUP COMMAND BUFFER, GET ATTENTION, SEND CMD
        RNZ                     ; RET IF ERROR
        LXI     H,HSTBUF        ;POINTER TO DATA SOURCE

        IF      H67BLKIO

WR671:  LDA     H67OUTD1        ;GET OUTPUT DATA PORT #
        MOV     C,A
WR672:  CALL    H67INS          ;GET THE BUS STATUS
        ANA     A               ;LOOK FOR REQ
        IF      HDBREQ-10000000B
%:      HDBREQ NE 10000000B
        ENDIF
        JP      WR672
        ANI     HDBCMD          ;CHECK FOR COMPLETION
        JNZ     CMPSTAT
        MVI     B,128           ;128 BYTE BLOCK
        DB      0EDH,0B3H       ;DO BLOCK OUTPUT
        JMP     WR672

        ELSE

WR671:  CALL    H67INS          ;GET THE BUS STATUS
        ANA     A               ;WAIT FOR REQ
        IF      HDBREQ-10000000B
%:      HDBREQ NE 10000000B
        ENDIF
        JP      WR671
        ANI     HDBCMD          ;CHECK FOR COMPLETION
        JNZ     CMPSTAT
        MOV     A,M             ;GET THE NEXT BYTE
```

```
        CALL    H67OUTD         ;SEND IT TO THE DISK
        INX     H
        JMP     WR671

        ENDIF

        PAGE
;
;  CHECK COMPLETION STATUS.
;
;    EXIT:  PSW/Z = 0 IF ERROR
;                 = 1 IF NO ERROR
;           'ERFLAG' = 0 IF NO ERROR
;                    <> 0 IF ERROR
;    USES:  ALL
;
CMPSTAT: CALL   H67IND          ;INPUT THE COMPLETION STATUS
        MOV     C,A             ;SAVE TEMPORARILY
LREQ:   CALL    H67INS          ;WAIT FOR LAST REQ
        ANA     A               ;REQ?
        IF      HDBREQ-10000000B
$:      HDBREQ NE 10000000B
        ENDIF
        JP      LREQ            ;NOT YET
        MOV     B,A             ;SAVE FOR CHECKING LATER
        CALL    H67IND          ;GET LAST BYTE

        EI                      ;ALLOW INTERRUPTS AGAIN

        ORA     A               ;SHOULD BE A BYTE OF ZEROS
        MVI     A,80H           ;ERROR CODE
        JNZ     H67ER           ; BR IF ERROR

        MOV     A,B
        ANI     HDBPE           ;CHECK FOR PARITY ERROR ON BUS
        MVI     A,81H           ;ERROR CODE
        JNZ     H67ER           ; BR IF ERROR

        MOV     A,C             ;GET THE COMPLETION BYTE
        ANI     HDFERR+HDFPE    ;CHECK COMPLETION STATUS
        RZ                      ; RET IF NO ERRORS

        ANI     HDFERR          ;CHECK IF I/O ERROR OR BAD PARITY
        MVI     A,82H           ;ERROR CODE FOR BAD PARITY
        JZ      H67ER           ; BR IF NOT I/O ERROR

        LDA     RS67B           ;CHECK IF ERROR DURING REQUEST SENSE
        ANA     A
        RNZ                     ; IF YES -- THEN DON'T TRY ANOTHER

        CALL    RS67            ;DO A REQUEST SENSE TO GET ERROR CODE

H67ER:  STA     ERRTYP          ;SAVE ERROR CODE

        CPI     HDEWF           ;CHECK FOR WRITE FAULT
```

```
        CZ      RCL67           ; IF YES -- RECALIBRATE HEAD TO CLEAR ERROR

        LXI     H,H67MSG        ;PRINT EXTENDED ERROR MESSAGE
        CALL    PRTERR

        ORI     OFFH            ;INDICATE ERROR
        STA     ERFLAG
        RET

H67MSG  DB      'H67',0

        PAGE
SETUP:  LXI     H,CMDBUF        ;POINTER TO THE COMMAND BUFFER
        MOV     M,A             ;SAVE COMMAND IN BUFFER STRING
        INX     H               ;POINT TO LOGICAL UNIT SLOT
        XCHG                    ;(DE)=COMMAND BUFFER POINTER
        LHLD    HSTDPB          ;GET UNIT SELECT VALUE
        INX     H
        MOV     A,M
        STAX    D               ;PLACE IT IN COMMAND BUFFER
        INX     D               ;BUMP COMMAND BUFFER POINTER
        PUSH    D               ;SAVE COMMAND BUFFER POINTER FOR LATER

;  COMPUTE LOGICAL SECTOR NUMBER.
        LHLD    HSTTRK          ;START WITH CP/M TRACK #
        MOV     D,H
        MOV     E,L
        DAD     H               ;*2
        DAD     D               ;*3
        DAD     H               ;*6
        DAD     H               ;*12
        DAD     D               ;*13
        DAD     H               ;*26
        IF      NSPT67-26
%:      NSPT67 NE 26
        ENDIF
        LDA     HSTSEC          ;SECTORS NUMBERED 0 TO SPT-1
        CALL    DADA            ;HL IS NOW THE LOGICAL SECTOR NUMBER
        XCHG                    ;NOW DE IS
        LHLD    HSTDPB          ;ADD TRACK 0 OFFSET
        LXI     B,DPETRK-DPEHTH
        DAD     B
        CALL    HLIHL           ;(HL) = TRACK 0 OFFSET
        XCHG                    ;(DE) = TRACK 0 OFFSET
                                ;(HL) = LOGICAL SECTOR #
        DAD     D

;  CHECK IF WITHIN PARTITION BOUNDARIES.
        MOV     A,D
        ORA     E
        JZ      SETUP2          ;BR IF FLOPPY OR NOT IN PARTITION MODE
                                ; (ASSUMED IF TRACK 0 OFFSET = 0)

        CALL    CPHLDE          ;CHECK LOGICAL SECTOR # AGAINST
                                ; BEGINNING OF PARTITION
        JC      SETUP9          ;  BR IF BEFORE
```

```
        XCHG                    ;(DE) = LOGICAL SECTOR #
        LHLD    HSTDPB          ;GET LAST SECTOR # + 1 OF PARTITION
        LXI     B,DPEUPB-DPEHTH
        DAD     B
        CALL    HLIHL
        XCHG                    ;(DE) = LAST SECTOR # + 1 OF PARTITION
                                ;(HL) = LOGICAL SECTOR #
        CALL    CPHLDE
        JNC     SETUP9          ; BR IF AFTER PARTITION

SETUP2: XCHG                    ;(DE) = LOGICAL SECTOR NUMBER

;  PUT LOGICAL SECTOR NUMBER INTO COMMAND BUFFER
        POP     H
        MOV     M,D             ;MSB FIRST
        INX     H
        MOV     M,E
        INX     H
        MVI     M,1             ;1 BLOCK
        INX     H
        MVI     M,0             ;0 THE CONTROL BYTE

;  SEND COMMAND TO H67 CONTROLLER.

SETUP3:
        XRA     A               ;INDICATE DO RS ON ERROR
        STA     RS67B

        LXI     D,-1            ;INIT TIMEOUT COUNTER

GETCON: CALL    H67INS          ;GET THE STATUS
        ANI     HDBBSY          ;IF NOT BUSY
        JZ      SETUP3A         ;  THEN GO AHEAD
        DCX     D
        MOV     A,D             ;CHECK TIMEOUT COUNTER
        ORA     E
        JZ      SETUP8          ; BR IF TIMEOUT
        JMP     GETCON

SETUP3A:
        MVI     A,HDFSEL        ;ASSERT SEL AND DATAO
        CALL    H67OUTC

        LXI     D,-1            ;INIT TIMEOUT COUNTER

CBUSY:  CALL    H67INS
        ANI     HDBBSY          ;IF BUSY
        JNZ     SETUP3B         ;  THEN WE GOT ITS ATTENTION
        DCX     D
        MOV     A,D             ;CHECK TIMEOUT COUNTER
        ORA     E
        JZ      SETUP8          ; BR IF TIMEOUT
        JMP     CBUSY

SETUP3B:
```

```
                DI                      ;DON'T WANT TO BE BOTHERED
                MVI     A,HDFDE         ;DATA ENABLE
                CALL    H67OUTC
                LXI     H,CMDBUF
        OUTCOM: CALL    H67INS
                ANI     HDBREQ+HDBCMD+HDBIO
                IF      HDBREQ-10000000B
        %:      HDBREQ NE 10000000B
                ENDIF
                JP      OUTCOM          ;WAIT FOR REQ
                CPI     HDBREQ+HDBCMD+HDBIO     ;CHECK FOR REQ/CMD/OUTPUT
                JNZ     SETUP4
                MOV     A,M             ;GET NEXT BYTE OF COMMAND
                CALL    H67OUTD
                INX     H
                JMP     OUTCOM
        SETUP4: XRA     A               ;INDICATE NO ERROR IN SETUP
                RET

        ;  ERROR IN SETUP DUE TO TIMEOUT.
        SETUP8: MVI     A,84H           ;ERROR CODE
                JMP     H67ER           ;HANDLE ERROR

        ;  ERROR IN SETUP DUE TO OUT OF BOUNDS CONDITION.
        SETUP9: POP     H               ;DISCARD COMMAND BUFFER POINTER
                MVI     A,83H           ;ERROR CODE
                JMP     H67ER           ;HANDLE ERROR

                PAGE
        ;
        ;  PRIMITIVE H67 I/O PORT ROUTINES.
        ;
        H67OUTD: OUT    0               ;OUTPUT TO DATA PORT
        H67OUTD1 EQU    $-1
                RET
        H67IND: IN      0               ;INPUT FROM DATA PORT
        H67IND1 EQU     $-1
                RET
        H67OUTC: OUT    0               ;OUTPUT TO CONTROL PORT
        H67OUTC1 EQU    $-1
                RET
        H67INS: IN      0               ;INPUT FROM BUS STATUS PORT
        H67INS1 EQU     $-1
                RET

        ;  H67 COMMAND BUFFER
        CMDBUF: DB      0               ;OPCODE
                DB      0               ;LUN | LOG ADR2
                DB      0               ;LOG ADR1
                DB      0               ;LOG ADR0
                DB      1               ;NUMBER OF BLOCKS
                DB      0               ;CONTROL BYTE

                ENDIF
                PAGE
```

```
07FA C30908   NULDVD: JMP    SETNUL
07FD C30908           JMP    RDNUL
0800 C30908           JMP    WRNUL
0803 C30908           JMP    RESNUL
0806 C30908           JMP    MNTNUL

                      SETNUL:
                      RDNUL:
                      WRNUL:
                      RESNUL:
                      MNTNUL:
0809 3EFF             MVI    A,0FFH
080B 322A0E           STA    ERFLAG
080E 37               STC
080F C9               RET

                      PAGE
```

```
                        ;
                        ;  2 MS CLOCK INTERRUPT SERVICE ROUTINE
                        ;
                                IF      TOD
                        NDAYS   DB      31,28,31,30,31,30,31,31,30,31,30,31
                                ENDIF
0810 0000000000TODVAL   DB      0,0,0,0,0,0             ;TIME OF DAY (SEC [0-59] ,
                                                        ; MIN [0-59] , HRS [0-23], DAY[1-N] ,
                                                        ; MON [1-12] , YR [0-255])
0816 0000       EVTCTR  DW      0                       ;EVENT DOWN COUNTER
0818 00         DLYMO:  DB      0
0819 00         DLYH:   DB      0
081A 00         DLYW:   DB      0

081B 22A70E     CLOCK:  SHLD    HSAV                    ;SAVE AF,HL
081E E1                 POP     H                       ;GET THE RETURN ADDRESS
081F 22A90E             SHLD    RETSAV                  ;SAVE IT, BUT NOT ON USER STACK
0822 F5                 PUSH    PSW                     ;SAVE AF, HL

0823 210D00             LXI     H,CTLPRT                ;GET THE CURRENT VALUE OF THE CONTROL PORT
0826 7E                 MOV     A,M
0827 D3F2               OUT     H88CTL                  ;AND OUTPUT AGAIN, RESETTING INT REQ

0829 23                 INX     H                       ;POINT TO THE H8FLAG
082A 7E                 MOV     A,M
082B B7                 ORA     A                       ;IF 0 THEN RUNNING IN H/Z89
082C CA3108             JZ      CLK0                    ;  THEN DON'T OUTPUT TO 360Q
082F D3F0               OUT     H8CTL                   ;  ELSE CONTAINS H8TR TO RESET H8 CLOCK

0831 2A0B00     CLK0:   LHLD    TICCNT                  ;GET THE TICK COUNTER
0834 23                 INX     H                       ;INCREMENT IT
0835 220B00             SHLD    TICCNT

0838 7D                 MOV     A,L                     ;IS IT A MULTIPLE OF 1/2 SECOND?
0839 B7                 ORA     A
083A C26708             JNZ     CLKRET                  ;  IF NOT

                        IF      TOD
                        MOV     A,H                     ;IS IT A MULTIPLE OF 1 SECOND?
                        RAR
                        JC      CLK1                    ;BR IF NOT
                        LXI     H,TODVAL                ;HANDLE TIME OF DAY
                        INR     M                       ;SECONDS
                        MOV     A,M
                        CPI     60
                        JC      CLK1
                        MVI     M,0
                        INX     H                       ; MINUTES
                        INR     M
                        MOV     A,M
                        CPI     60
                        JC      CLK1
                        MVI     M,0
                        INX     H                       ; HOURS
                        INR     M
```

```
                        MOV     A,M
                        CPI     24
                        JC      CLK1
                        MVI     M,0
                        INX     H               ; DAYS
                        INX     H
                        MOV     A,M
                        LXI     H,NDAYS-1
                        ADD     L
                        MOV     L,A
                        MOV     A,H
                        ACI     0
                        MOV     H,A
                        MOV     A,M
                        LXI     H,TODVAL+3
                        INR     M
                        CMP     M
                        JNC     CLK1
                        MVI     M,1
                        INX     H               ; MONTHS
                        INR     M
                        MOV     A,M
                        CPI     13
                        JC      CLK1
                        MVI     M,1
                        INX     H               ; YEARS
                        INR     M
                        ENDIF

                CLK1:

                        IF      EVENT
                        LHLD    EVTCTR          ;DOWN COUNT EVENT COUNTER IF <> 0
                        MOV     A,H
                        ORA     L
                        JZ      CLK2
                        DCX     H
                        SHLD    EVTCTR
                        ENDIF

                CLK2:

                        IF      H17T
083D 211808             LXI     H,DLYMO         ;POINTER TO MOTOR DELAY TIMER
0840 7E                 MOV     A,M
0841 B7                 ORA     A               ;IF ALREADY ZERO
0842 CA6708             JZ      CLK4            ;  THEN DON'T DECREMENT

0845 35                 DCR     M               ;DECREMENT TIMER
0846 C25308             JNZ     CLK3            ;  IF IT HAS NOT TIMED OUT CHECK HEADS

0849 3A0F00             LDA     DEVCTL          ;GET THE CURRENT VALUE OF CONTROL PORT
084C E6EF               ANI     0FFH-DFMO       ;TURN OFF MOTOR
084E 320F00             STA     DEVCTL
0851 D37F               OUT     DPDC
```

```
0853 23          CLK3:  INX   H          ;POINT TO THE HEAD DELAY
0854 7E                 MOV   A,M
0855 B7                 ORA   A          ;IF ALREADY ZERO
0856 CA6708             JZ    CLK4       ;  THEN DON'T DECREMENT

0859 35                 DCR   M          ;DECREMENT TIMER
085A C26708             JNZ   CLK4       ;  IF IT HAS NOT TIMED OUT THEN SKIP

085D 3A0F00             LDA   DEVCTL     ;DESELECT THE DRIVE
0860 E6F1               ANI   OFFH-U0-U1-U2
0862 320F00             STA   DEVCTL
0865 D37F               OUT   DPDC
                        ENDIF

                 CLK4:
                        IF    H37T
                        LXI   H,DLYMO37  ;POINTER TO MOTOR DELAY TIME FOR H37
                        MOV   A,M
                        ORA   A          ;IF ALREADY ZERO
                        JZ    CLKRET     ;  THEN DON'T DECREMENT

                        DCR   M          ;DECREMENT TIMER
                        JNZ   CLK5       ;  IF IT HAS NOT TIMED OUT CHECK HEADS

                        LDA   H37CTL     ;GET THE CURRENT VALUE OF CONTROL PORT
                        ANI   OFFH-CONMO ;TURN OFF MOTOR
                        STA   H37CTL
                        OUT   FD$CON

                 CLK5:  INX   H          ;POINT TO THE HEAD DELAY FOR H37
                        MOV   A,M
                        ORA   A          ;IF ALREADY ZERO
                        JZ    CLKRET     ;  THEN DON'T DECREMENT

                        DCR   M          ;DECREMENT TIMER
                        JNZ   CLKRET     ;  IF IT HAS NOT TIMED OUT THEN SKIP

                        LDA   H37CTL     ;DESELECT THE DRIVE
                        ANI   OFFH-CONDS0-CONDS1-CONDS2-CONDS3
                        STA   H37CTL
                        OUT   FD$CON
                        ENDIF

0867 3A0B00      CLKRET: LDA  TICCNT
086A 1F                 RAR              ;IS IT EVEN, MAKING 4MS BIG TICKS
086B DA7708             JC    CLKR2
086E 211A08             LXI   H,DLYW     ;CHECK WAIT TIMER
0871 7E                 MOV   A,M        ;  AND DECREMENT IT IF IT IS NOT
0872 B7                 ORA   A          ;  ALREADY ZERO
0873 CA7708             JZ    CLKR2
0876 35                 DCR   M
0877 F1          CLKR2: POP   PSW        ;RESTORE THE MACHINE STATE
0878 2AA90E             LHLD  RETSAV
087B E5                 PUSH  H
087C 2AA70E             LHLD  HSAV
087F FB                 EI
```

```
0880 C9          RET
                 PAGE
```

```
                        ;
                        ; CHKLAB -- CHECK CHECKSUM OF LABEL
                        ;
                        ;    ENTRY:  'HSTBUF' CONTAINS SECTOR WITH LABEL
                        ;    EXIT:   PSW/Z = 0 IF BAD CHECKSUM
                        ;                  = 1 IF GOOD CHECKSUM
                        ;    USES:   A,F,B,H,L
                        ;

                        CHKLAB:
0881 AF                 XRA     A               ;ZERO ACCUM
0882 0619               MVI     B,LABLEN        ;GET LENGTH OF LABEL
0884 21380C             LXI     H,HSTBUF+LABEL

                        CHKLAB1:
0887 86                 ADD     M               ;ADD VALUES
0888 23                 INX     H
0889 05                 DCR     B
088A C28708             JNZ     CHKLAB1

088D 3C                 INR     A               ;INR CHECKSUM VALUE AND SET/RESET PSW/Z

088E C9                 RET

                        ;
                        ; CPHLDE - COMPARE (HL) TO (DE)
                        ;    USES A,F
                        ;

                        CPHLDE:
088F 7C                 MOV     A,H
0890 BA                 CMP     D
0891 C0                 RNZ
0892 7D                 MOV     A,L
0893 BB                 CMP     E
0894 C9                 RET
                        ;
                        ; DADA -  ADD 0,A  TO HL
                        ;       USES AF

0895 85                 DADA:   ADD     L
0896 6F                 MOV     L,A
0897 D0                 RNC
0898 24                 INR     H
0899 C9                 RET
                        ;
                        ;  GETDPE -- GET ADDRESS OF DPE
                        ;
                        ;    ENTRY:  (A) = LOGICAL/MAPPED DRIVE #
                        ;    EXIT:   (HL) = ADDRESS OF DPE
                        ;    USES:   A,F,D,E,H,L
                        ;

                        GETDPE:
089A E60F               ANI     0FH             ;GET MAPPED DRIVE #
```

```
089C 6F                 MOV     L,A             ;FIND DPE ADDR

089D 2600               MVI     H,0
089F 54                 MOV     D,H
08A0 5D                 MOV     E,L
08A1 29                 DAD     H               ;*2
08A2 19                 DAD     D               ;*3
08A3 29                 DAD     H               ;*6
08A4 29                 DAD     H               ;*12
08A5 29                 DAD     H               ;*24
                        IF      DPEL-24
                %:      DPEL NE 24
                        ENDIF

08A6 114E00             LXI     D,DPBASE
08A9 19                 DAD     D

08AA C9                 RET

                ;
                ;  GETDPEX -- GET ADDR OF DPE'S HEATH EXTENSIONS
                ;
                ;    ENTRY:  (A) = LOGICAL/MAPPED DRIVE #
                ;    EXIT:   (HL) = ADDR OF HEATH EXTENSIONS
                ;    USES:  A,F,D,E,H,L
                ;

                GETDPEX:
08AB CD9A08             CALL    GETDPE          ;GET ADDR OF DPE

08AE 111000             LXI     D,DPEHTH        ;GET ADDR OF HEATH EXTENSIONS
08B1 19                 DAD     D

08B2 C9                 RET

                ;
                ; HLIHL - LOAD HL INDIRECT THROUGH HL
                ;       USES AF

08B3 7E         HLIHL:  MOV     A,M
08B4 23                 INX     H
08B5 66                 MOV     H,M
08B6 6F                 MOV     L,A
08B7 C9                 RET

                ;
                ;  MOVEITX -- MOVE DATA FROM ONE AREA OF MEMORY TO ANOTHER
                ;
                ;    ENTRY:   (C) = COUNT
                ;            (DE) = SOURCE
                ;            (HL) = DESTINATION
                ;    USES:   ALL
                ;

                MOVEITX:
08B8 EB                 XCHG                    ;(HL)=SOURCE  (DE)=DESTINATION
```

```
                        IF      $-MOVEIT
                $:      MOVEIT MUST IMMEDIATELY FOLLOW MOVEITX
                        ENDIF

                ;
                ;  MOVEIT -- MOVE DATA FROM ONE AREA OF MEMORY TO ANOTHER
                ;
                ;    ENTRY:   (C) = COUNT
                ;            (DE) = DESTINATION
                ;            (HL) = SOURCE
                ;    USES:    ALL
                ;

08B9 3E81       MOVEIT: MVI     A,81H
08BB C680               ADI     80H
08BD E2C508             JPO     MOVEIT1         ;BR IF RUNNING ON 8080

08C0 0600               MVI     B,0
08C2 EDB0               DB      0EDH,0B0H       ;USE Z80 BLOCK MOVE INSTRUCTION
08C4 C9                 RET

                MOVEIT1:
08C5 7E                 MOV     A,M             ;MOVE DATA USING 8080 CODE
08C6 12                 STAX    D
08C7 23                 INX     H
08C8 13                 INX     D
08C9 0D                 DCR     C
08CA C2C508             JNZ     MOVEIT1
08CD C9                 RET

                        PAGE
```

```
                        ;*********************************************
                        ;
                        ;       LOGICAL DEVICE ROUTINES
                        ;
                        ;       THESE ROUTINES HANDLE THE LOGICAL TO PHYSICAL
                        ;       DEVICE MAPPING ESTABLISHED BY THE CP/M IOBYTE
                        ;

                        ;
                        ; CONSOLE STATUS
                        ;

08CE CDD608             CONST:  CALL    CONS            ;GET STATUS OF SPECIFIC DEVICE
08D1 B7                         ORA     A
08D2 C8                         RZ                      ;IF NOT READY RETURN 0 IN A
08D3 3EFF                       MVI     A,0FFH          ; ELSE RETURN FF
08D5 C9                         RET
                        ;
08D6 3A0300             CONS:   LDA     IOBYTE          ;USE BITS 1-0 FOR CONSOLE DEVICE
08D9 CD4F09                     CALL    INDXIT
08DC 310A                       DW      TTYSTAT
08DE 5B09                       DW      CRTSTAT
08E0 E408                       DW      RDRST           ;2: BATCH MODE (USE READER DEVICE)
08E2 5B09                       DW      CRTSTAT
                        ;
                        ; READER STATUS

08E4 3A0300             RDRST:  LDA     IOBYTE
08E7 0F                         RRC
08E8 CD5009                     CALL    GOTOIT
08EB 310A                       DW      TTYSTAT
08ED 500B                       DW      BUSY            ;UNIMPLEMENTED INPUTS
08EF 830A                       DW      MDSTAT
08F1 5B09                       DW      CRTSTAT
                        ;
                        ; CONSOLE INPUT
                        ;
08F3 3A0300             CONIN:  LDA     IOBYTE
08F6 CD4F09                     CALL    INDXIT
08F9 280A                       DW      TTYIN           ;0: TTY
08FB 7C09                       DW      CRTIN           ;1: CRT
08FD 4009                       DW      READER          ;2: BAT (READER INPUT)
08FF 7C09                       DW      CRTIN           ;UC1: CRT INPUT, LST: OUTPUT
                        ;
                        ; CONSOLE OUT
                        ;
0901 3A0300             CONOUT: LDA     IOBYTE
0904 CD4F09                     CALL    INDXIT
0907 370A                       DW      TTYOUT          ;0: TTY
0909 080A                       DW      CRTOUT          ;1: CRT
090B 1F09                       DW      LIST            ;2: BAT (OUTPUT TO LST)
090D 1F09                       DW      LIST            ;UC1: CRT INPUT, LST: OUTPUT
                        ;
                        ; LISTST - LIST STATUS CHECK
                        ;
```

```
090F 3A0300   LISTST: LDA    IOBYTE        ;GET THE CURRENT IOBYTE
0912 07               RLC                  ;SHIFT INTO POSITION
0913 07               RLC
0914 CD4F09           CALL   INDXIT
0917 9C0A             DW     TTYOS         ;0: TTY
0919 A50A             DW     CRTOS         ;1: CRT
091B DF0A             DW     LPTOS         ;2: LPT
091D 180B             DW     DBDOS         ;3: DIABLO

                      ;
                      ; LIST OUT
                      ;

091F 3A0300   LIST:   LDA    IOBYTE
0922 07               RLC                  ;BITS 7-6 TO 2-1
0923 07               RLC
0924 CD4F09           CALL   INDXIT
0927 370A             DW     TTYOUT        ;0: TTY
0929 080A             DW     CRTOUT        ;1: CRT
092B 410A             DW     LPTOUT        ;2: LPT
092D 5C0A             DW     DBD           ;3: DIABLO
                      ;
                      ; PUNCH OUT
                      ;

092F 3A0300   PUNCH:  LDA    IOBYTE        ;BITS 4-5 TO 1-2
0932 0F               RRC
0933 0F               RRC
0934 0F               RRC
0935 CD5009           CALL   GOTOIT
0938 370A             DW     TTYOUT        ;0: TTY
093A 540B             DW     DMYOUT
093C 890A             DW     MDOUT         ;2: UP1 MODEM PORT OUTPUT
093E 080A             DW     CRTOUT
                      ;
                      ; READER IN
                      ;

0940 3A0300   READER: LDA    IOBYTE        ;BITS 3-2 TO 2-1
0943 0F               RRC
0944 CD5009           CALL   GOTOIT
0947 280A             DW     TTYIN         ;0: TTY
0949 520B             DW     DMYIN
094B 7D0A             DW     MDIN          ;2: UR1 MODEM PORT INPUT
094D 7C09             DW     CRTIN
                      ;
                      ; DISPATCH SUBROUTINE - INDEXED TABLE JUMP
                      ;
094F 07       INDXIT: RLC
0950 E606     GOTOIT: ANI    06H           ;MASK BITS
0952 E3               XTHL                 ;SAVE HL, GET TABLE ADDRESS
0953 CD9508           CALL   DADA          ;ADD 0,A TO HL
0956 CDB308           CALL   HLIHL         ;GET ADDRESS IN HL
0959 E3               XTHL                 ;XCHG ROUTINE ADDRESS, OLD HL
095A C9               RET                  ;DISPATCH
```

PAGE

```
                        ;*****************************************************
                        ;
                        ;       PHYSICAL DEVICE ROUTINES
                        ;
                        ; ACCESSED VIA THE LOGICAL DEVICE ROUTINES ABOVE
                        ;

                        ;
                        ;"CRT" PHYSICAL STATUS ROUTINE
                        ;       USES H84PT1

                        CRTSTAT:
                                IF      NOT INTINP
                                LDA     MODE            ;GET THE MODE BYTE
                                RAR                     ;IF THE LSB = 1
                                JC      CRTS1           ;  THEN CONSOLE ON H8-5
                                IF      MODEBO-1
                        %:      MODEBO NE 1
                                ENDIF

                                LXI     H,H84PT1        ;POINTER TO BASE PORT
                                JMP     US              ;GET STATUS

                        CRTS1:  IN      H85CRT+1        ;GET 8251 STATUS REGISTER
                                ANI     02H             ;MASK RXRDY
                                RET
                                ENDIF

                                IF      INTINP
095B 3A090C                     LDA     CRTB            ;FIND OUT HOW MANY CHARACTERS ARE IN THE BUFFER
095E B7                         ORA     A
095F C0                         RNZ                     ;RET IF CHARACTER AVAILABLE

                        ;  INSURE INTERRUPTS FOR CRT INPUT ARE ENABLED.
                        ;  A USER PROGRAM MAY HAVE TURN THEM OFF.
0960 3A3600             CRTS2:  LDA     MODE            ;Q. CRT ON H8-5 CARD
0963 1F                         RAR
0964 D26E09                     JNC     CRTS2A          ; BR IF NOT
                                IF      MODEBO-1
                        %:      MODEBO NE 1
                                ENDIF

0967 3E17                       MVI     A,17H           ;ENABLE RX & TX PLUS INTERRUPTS
0969 D3FB                       OUT     H85CRT+1
096B C37909                     JMP     CRTS3

096E 3A3700             CRTS2A: LDA     H84PT1          ;ENABLE RECEIVER INTERRUPTS ON 8250
0971 3C                         INR     A
0972 327809                     STA     CRTS2B
0975 3E01                       MVI     A,1
0977 D300                       OUT     0
0978 =                  CRTS2B  EQU     $-1

0979 FB                 CRTS3:  EI                      ;INSURE MASTER ENABLE
```

```
097A AF                 XRA     A               ;INDICATE NO CHARACTER AVAILABLE
097B C9                 RET
                        ENDIF

                ;
                ;"CRT" PHYSICAL INPUT ROUTINE
                ;

                CRTIN:
                        IF      NOT INTINP
                        LDA     MODE            ;GET MODE BYTE
                        RAR                     ;IF LSB = 1
                        JC      CRTI1           ;  THEN CONSOLE ON H8-5
                        IF      MODEBO-1
                %:      MODEBO NE 1
                        ENDIF

                        LXI     H,H84PT1
                        CALL    UI              ;GET CHAR FROM 8250
                        ANI     7FH             ;MASK PARITY
                        RET

                CRTI1:  IN      H85CRT+1        ;CHECK IF RXRDY
                        ANI     02H
                        JZ      CRTI1           ;WAIT FOR CHARACTER

                        IN      H85CRT          ;GET CHARACTER
                        ANI     7FH             ;MASK PARITY
                        RET
                        ENDIF

                        IF      INTINP
097C CD5B09     CRTIN1: CALL    CRTSTAT         ;CHECK IF CHARACTER AVAILABLE
097F CA7C09             JZ      CRTIN1          ;BR IF NO CHARACTER AVAILABLE

0982 F3                 DI                      ;A CHARACTER IS AVAILABLE -- DON'T
                                                ; LET ANYONE BOTHER ME WHILST I GET IT
0983 21090C             LXI     H,CRTB          ;POINTER TO NUMBER OF CHARS IN BUFFER
0986 35                 DCR     M               ;MARK ONE TAKEN
0987 2A0A0C             LHLD    CRTGET          ;GET THE POINTER TO THE CHAR
098A 4E                 MOV     C,M             ;PUT THE CHAR IN C
098B 23                 INX     H               ;ADVANCE THE POINTER
098C 7D                 MOV     A,L             ;CHECK THAT IT WASN'T ADVANCED PAST END
098D FED5               CPI     CRTBND MOD 256
098F C29509             JNZ     CRTIN2
0992 21AD0E             LXI     H,CRTBUF
0995 220A0C     CRTIN2: SHLD    CRTGET
0998 FB                 EI                      ;ALLOW ME TO BE BOTHERED
0999 79                 MOV     A,C             ;PUT THE CHARACTER IN A (TOO)
099A C9                 RET

                ; CRT INTERRUPT SERVICE ROUTINE

099B 22A70E     CRTISR: SHLD    HSAV            ;SAVE THE PROCESSOR STATE
099E E1                 POP     H               ;GET RETURN ADDRESS
099F 22A90E             SHLD    RETSAV
```

```
09A2 F5                 PUSH    PSW
09A3 210000             LXI     H,0             ;SAVE THE OLD SP
09A6 39                 DAD     SP
09A7 22AB0E             SHLD    OLDSP
09AA 31E50E             LXI     SP,LCLSTK       ;SET UP OUR VERY OWN STACK
09AD 213700             LXI     H,H84PT1        ;POINTER TO SERIAL DEVICE STRUCTURE
09B0 3E02               MVI     A,2             ;INPUT INTERRUPT IDENTIFICATION REG
09B2 CD8D0B             CALL    PINX
09B5 FE04               CPI     0100B           ;CHECK FOR RECEIVED DATA AVAILABLE INT
09B7 C2010A             JNZ     CRTIS6          ;IT WASN'T THIS 8250

                        IF      BRKKEY
                        MVI     A,5             ;CHECK FOR BREAK
                        CALL    PINX
                        ANI     10H
                        JNZ     CRTIS8
                        ENDIF

09BA CD8C0B             CALL    UI1             ;GET THE CHARACTER

                CRTIS1:
09BD E67F               ANI     7FH             ;MASK PARITY BIT
09BF F5                 PUSH    PSW             ;SAVE THE CHARACTER
09C0 3A090C             LDA     CRTB            ;GETTING NEAR THE END OF THE BUFFER?
09C3 FE24               CPI     CRTLEN-4
09C5 DAD909             JC      CRTIS2          ;NOT YET
09C8 C5                 PUSH    B
09C9 D5                 PUSH    D
09CA 0E07               MVI     C,BELL          ;WARN HIM BY SENDING BELL
09CC CD080A             CALL    CRTOUT
09CF D1                 POP     D
09D0 C1                 POP     B
09D1 3A090C             LDA     CRTB            ;CAN WE ACCEPT THIS CHARACTER
09D4 FE28               CPI     CRTLEN
09D6 CAFD09             JZ      CRTIS5

09D9 F1         CRTIS2: POP     PSW             ;RECALL  THE CHARACTER
09DA 2A0C0C             LHLD    CRTPUT          ;THE BUFFER PUT POINTER
09DD 77                 MOV     M,A             ;PUT THE CHARACTER IN THE BUFFER
09DE 23                 INX     H               ;ADVANCE THE POINTER
09DF 7D                 MOV     A,L             ;CHECK FOR WRAP-AROUND
09E0 FED5               CPI     CRTBND MOD 256
09E2 C2E809             JNZ     CRTIS3
09E5 21AD0E             LXI     H,CRTBUF
09E8 220C0C     CRTIS3: SHLD    CRTPUT
09EB 21090C             LXI     H,CRTB
09EE 34                 INR     M               ;ANOTHER CHARACTER IS AVAILABLE

09EF 2AAB0E     CRTIS4: LHLD    OLDSP           ;RESTORE THE OLD STACK
09F2 F9                 SPHL
09F3 F1                 POP     PSW             ;RESTORE THE MACHINE STATE
09F4 2AA90E             LHLD    RETSAV
09F7 E5                 PUSH    H
09F8 2AA70E             LHLD    HSAV
09FB FB                 EI                      ;ENABLE MORE INTERRUPTS
09FC C9                 RET                     ;AND RETURN
```

```
09FD F1          CRTIS5: POP     PSW             ;CLEAN STACK
09FE C3EF09              JMP     CRTIS4

0A01 DBFB        CRTIS6: IN      H85CRT+1        ;GET STATUS
                         IF      BRKKEY
                         ANI     20H             ;HAD AN OVERRUN?
                         JNZ     CRTI11          ;  YES, BREAKOUT
                         ENDIF

0A03 DBFA                IN      H85CRT          ;IF THE CRT 8250 DIDN'T DO IT, THE 8251 DID
0A05 C3BD09              JMP     CRTIS1

                         IF      BRKKEY
                 CRTIS8: CALL    UI1             ;GET THE GARBAGE
                         LXI     D,6000          ;WAIT FOR THE DUST TO SETTLE
                 CRTIS9: MVI     A,5
                         CALL    PINX
                         ANI     9
                         JNZ     CRTIS8
                         DCX     D
                         MOV     A,D
                         ORA     E
                         JNZ     CRTIS9
                         CALL    UI1             ;MAKE SURE THERE IS NO GARBAGE PRESENT

                 CRTI10: LXI     H,CRTBUF
                         SHLD    CRTGET
                         SHLD    CRTPUT
                         XRA     A
                         STA     CRTB
                         IF      H17T
                         CALL    RESH17
                         ENDIF
                         IF      H37T
                         CALL    RESH37
                         ENDIF
                         IF      H47T
                         CALL    RESH47
                         ENDIF
                         IF      H67T
                         CALL    RESH67
                         ENDIF
                         CALL    FLUSH1          ;FLUSH (ABORT) HOST BUFFER
                         EI
                         JMP     BOOT

                 CRTI11: IN      H85CRT
                         MVI     A,17H
                         OUT     H85CRT+1

                         LXI     D,6000
                 CRTI12: IN      H85CRT+1
                         ANI     22H
                         JNZ     CRTI11
                         DCX     D
                         MOV     A,D
```

```
                        ORA     E
                        JNZ     CRTI12

                        IN      H85CRT
                        JMP     CRTI10
                        ENDIF

                        ENDIF
                        ;
                        ;"CRT" PHYSICAL OUTPUT ROUTINES
                        ;

0A08 CDA50A     CRTOUT: CALL    CRTOS
0A0B B7                 ORA     A
0A0C CA080A             JZ      CRTOUT

0A0F 3A3600             LDA     MODE            ;GET MODE BYTE
0A12 1F                 RAR                     ;IF LSB = 1
0A13 DA190A             JC      CRTO1           ;  THEN CONSOLE ON H8-5
                        IF      MODEBO-1
                        %:      MODEBO NE 1
                        ENDIF

0A16 C3650B             JMP     UO              ;OUTPUT CHARACTER IN C

0A19 E5         CRTO1:  PUSH    H
0A1A 23                 INX     H
0A1B 23                 INX     H               ;POINT TO FLAG BYTE
0A1C 7E                 MOV     A,M
0A1D 17                 RAL
0A1E E1                 POP     H
0A1F 79                 MOV     A,C
0A20 DC940B             CC      MUC             ;MAP TO UPPER CASE
0A23 D3FA               OUT     H85CRT

0A25 C3750B             JMP     POUT2           ;CHECK FOR NULLS

                        ;
                        ; TTY INPUT
                        ;

0A28 213A00     TTYIN:  LXI     H,H84PT2
0A2B CD860B             CALL    UI
0A2E E67F               ANI     7FH             ;MASK PARITY
0A30 C9                 RET
                        ;
                        ; TTY STATUS
                        ;

0A31 213A00     TTYSTAT:LXI     H,H84PT2
0A34 C3550B             JMP     US              ;GET STATUS

                        ;
                        ; TTY OUTPUT
                        ;
```

```
0A37 CD9C0A    TTYOUT: CALL    TTYOS
0A3A B7                ORA     A
0A3B CA370A            JZ      TTYOUT
0A3E C3650B            JMP     UO              ;OUTPUT CHARACTER IN C

                       ;
                       ; LINE PRINTER OUT
                       ;

0A41 3A030C    LPTOUT: LDA     DCLPOS          ;IF DON'T CHECK LP OUTPUT STATUS
0A44 B7                ORA     A
0A45 C24F0A            JNZ     LPTOU2          ;  THEN SKIP THE TEST

0A48 CDDF0A    LPTOU1: CALL    LPTOS           ;  ELSE, WAIT FOR READY LP OUTPUT STATUS
0A4B B7                ORA     A
0A4C CA480A            JZ      LPTOU1

0A4F 213D00    LPTOU2: LXI     H,H84PT3        ;POINTER TO DEVICE STRUCTURE
0A52 11060C            LXI     D,LPTCTS        ;AND ONE TO CHAR TO SEND

0A55 AF                XRA     A               ;FORCE A CHECK OF LP OUTPUT STATUS
0A56 32030C            STA     DCLPOS          ;  NEXT TIME

0A59 C3650B            JMP     UO

                       ;
                       ; DIABLO ETX/ACK PROTOCOL DRIVER
                       ;

0A5C CD180B    DBD:    CALL    DBDOS
0A5F B7                ORA     A
0A60 CA5C0A            JZ      DBD
0A63 CD650B            CALL    UO              ;SEND CHARACTER IN C TO PRINTER

0A66 217C0A            LXI     H,HSCNT         ;UPDATE HANDSHAKE COUNT
0A69 35                DCR     M
0A6A FE1B              CPI     01BH            ;ESC?
0A6C 7E                MOV     A,M
0A6D C2760A            JNZ     DBD1            ;WAS NOT AN ESCAPE
0A70 FE02              CPI     2               ;LAST CHAR WAS ESCAPE,
0A72 D0                RNC                     ;  MAKE CERTAIN AT LEAST TWO CHARS FOLLOW
0A73 3602              MVI     M,2             ;  WITHOUT INTERVENING ETX
0A75 C9                RET
0A76 B7        DBD1:   ORA     A               ;TIME TO HANDSHAKE?
0A77 C0                RNZ
0A78 3E01              MVI     A,1             ;TELL DBDOS IT IS TIME TO HANDSHAKE
0A7A 12                STAX    D
0A7B C9                RET

0A7C 20        HSCNT:  DB      32

                       ;
                       ; MDIN - MODEM INPUT ROUTINE
                       ;

0A7D 214000    MDIN:   LXI     H,H84PT4
```

```
0A80 C3860B             JMP     UI

                        ;
                        ; MDSTAT - MODEM INPUT STATUS
                        ;

0A83 214000     MDSTAT: LXI     H,H84PT4
0A86 C3550B             JMP     US

                        ;
                        ; MDOUT - MODEM OUTPUT
                        ;

0A89 CD930A     MDOUT:  CALL    MDOS
0A8C B7                 ORA     A
0A8D CA890A             JZ      MDOUT
0A90 C3650B             JMP     UO

                        ;
                        ; MDOS, TTYOS, AND CRTOS - MODEM, TTY, AND CRT OUTPUT STATUS
                        ;       RETURNS 00 FOR BUSY
                        ;               FF FOR READY TO ACCEPT ANOTHER CHARACTER

0A93 214000     MDOS:   LXI     H,H84PT4
0A96 11070C             LXI     D,MDCTS
0A99 C3B20A             JMP     CRTOS1

0A9C 213A00     TTYOS:  LXI     H,H84PT2
0A9F 11040C             LXI     D,TTYCTS
0AA2 C3B20A             JMP     CRTOS1

0AA5 213700     CRTOS:  LXI     H,H84PT1
0AA8 11050C             LXI     D,CRTCTS

0AAB 3A3600             LDA     MODE            ;HANDLE H8-5 CASE SPECIALLY
0AAE 1F                 RAR
0AAF DACA0A             JC      CRTOS3

0AB2 CD5D0B     CRTOS1: CALL    UOS     ;CHECK TO SEE IF THE UART CAN TAKE A CHAR
0AB5 CAC80A             JZ      CRTOSB          ;  THEN RETURN FLAGGING BUSY

0AB8 1A                 LDAX    D               ;SEE IF THERE ARE ANY NULLS TO BE SENT
0AB9 B7                 ORA     A
0ABA C2BF0A             JNZ     CRTOS2          ;IF SO, GO SEND ONE

0ABD 3D                 DCR     A               ;ELSE, SET READY
0ABE C9                 RET

0ABF 3D         CRTOS2: DCR     A               ;COUNT THIS NULL AS SENT
0AC0 12                 STAX    D
0AC1 C5                 PUSH    B

0AC2 0E00               MVI     C,NULL          ;SEND A NULL
0AC4 CD650B             CALL    UO

0AC7 C1                 POP     B
```

```
0AC8 AF          CRTOSB: XRA     A              ;RETURN CLAIMING TO BE STILL BUSY
0AC9 C9                  RET

0ACA DBFB        CRTOS3: IN      H85CRT+1       ;SPECIAL CASE: H8-5 SERIAL CARD
0ACC 1F                  RAR
0ACD D2C80A              JNC     CRTOSB         ;STILL BUSY

0AD0 1A                  LDAX    D              ;ANY NULLS STILL TO SEND?
0AD1 B7                  ORA     A
0AD2 C2D70A              JNZ     CRTOS4         ;IF SO, GO SEND ONE

0AD5 3D                  DCR     A              ;RETURN READY
0AD6 C9                  RET

0AD7 3D          CRTOS4: DCR     A              ;COUNT THIS NULL AS SENT
0AD8 12                  STAX    D

0AD9 3E00                MVI     A,NULL         ;SEND A NULL
0ADB D3FA                OUT     H85CRT

0ADD AF                  XRA     A              ;RETURN BUSY
0ADE C9                  RET

                 ;
                 ; LPTOS - LINE PRINTER OUTPUT STATUS
                 ;        WITH HARDWARE HANDSHAKE
                 ;

0ADF 213D00      LPTOS:  LXI     H,H84PT3
0AE2 11060C              LXI     D,LPTCTS

0AE5 CD5D0B              CALL    UOS     ;CHECK TO SEE IF THE UART CAN TAKE A CHAR
0AE8 CA160B              JZ      LPTOSB         ;THE UART IS STILL BUSY

                 ;       HANDSHAKE USED FOR H14/WH24
                 ;           IF YOUR PRINTER DOES NOT USE HANDSHAKE TO INDICATE "BUSY"
                 ;           DELETE THE NEXT 5 LINES

0AEB 3E06                MVI     A,6            ;INPUT MODEM STATUS REG
0AED CD8D0B              CALL    PINX
0AF0 47                  MOV     B,A
0AF1 3A3600              LDA     MODE           ;CHECK FOR POLARITY OF READY SIGNAL
0AF4 E604                ANI     MODEB2
0AF6 C2FC0A              JNZ     LPTOS0         ; BR IF READY IS HIGH POLARITY
0AF9 78                  MOV     A,B            ;READY IS INDICATED BY LOW POLARITY
0AFA 2F                  CMA                    ; THEREFORE, COMPLEMENT STATUS
0AFB 47                  MOV     B,A            ;  BEFORE CHECKING
0AFC 3A3500      LPTOS0: LDA     PRTRDY         ;GET LPT PRINTER READY MASK
0AFF A0                  ANA     B              ;CHECK APPROPRIATE READY LINE
0B00 CA160B              JZ      LPTOSB         ; BR IF NOT READY

0B03 1A                  LDAX    D              ;ANY NULLS TO SEND?
0B04 B7                  ORA     A
0B05 C20D0B              JNZ     LPTOS1         ;YES, THERE ARE NULLS REQUIRED
```

```
0B08 3D                 DCR     A               ;NO, RETURN WITH A = OFFH INDICATING READY
0B09 32030C             STA     DCLPOS          ;FLAG DON'T CHECK LP STATUS
0B0C C9                 RET

0B0D 3D         LPTOS1: DCR     A               ;COUNT THIS NULL AS SENT
0B0E 12                 STAX    D

0B0F C5                 PUSH    B               ;SAVE THE ORIGINAL CHARACTER

0B10 0E00               MVI     C,NULL
0B12 CD650B             CALL    UO

0B15 C1                 POP     B

0B16 AF         LPTOSB: XRA     A               ;INDICATE BUSY
0B17 C9                 RET

                ;
                ; DBDOS - DIABLO OUTPUT STATUS
                ;       IF CTS == 0 THEN OKAY TO SEND CHARACTERS
                ;          CTS == 1 THEN SEND ETX, SET CTS TO 2
                ;          CTS == 2 THEN WAIT FOR ACK, THEN SET CTS TO 0

0B18 213D00     DBDOS:  LXI     H,H84PT3
0B1B 11080C             LXI     D,DBDCTS

0B1E 1A                 LDAX    D               ;FIND OUT THE STATE OF OUTPUT
0B1F FE02               CPI     2               ;IF NOT 2,
0B21 C23A0B             JNZ     DBDOS1          ;  THEN GO DO OUTPUT

                ; MUST RECEIVE AN ACK FROM THE PRINTER

0B24 CD550B             CALL    US              ;CHECK UART FOR INCOMING
0B27 CA500B             JZ      DBDOSB          ;NO CHARACTER BACK FROM PRINTER YET
                                                ;  SO FLAG BUSY
0B2A CD8C0B             CALL    UI1             ;GET THE CHARACTER
0B2D E67F               ANI     07FH            ;STRIP OFF PARITY
0B2F D606               SUI     'F' MOD 32      ;COMPARE IT TO ACK
0B31 C2500B             JNZ     DBDOSB          ;NOT AN ACK, SO STILL BUSY
0B34 12                 STAX    D               ;WAS AN ACK, SO ABLE TO SEND MORE CHARS

0B35 3E20               MVI     A,32            ;RESET THE HANDSHAKE COUNT
0B37 327C0A             STA     HSCNT

0B3A CD5D0B     DBDOS1: CALL    UOS     ;CHECK TO SEE IF UART CAN TAKE A CHAR
0B3D CA500B             JZ      DBDOSB          ;UART IS NOT READY TO ACCEPT A CHARACTER

0B40 1A                 LDAX    D               ;IS IT TIME TO SEND ETX?
0B41 B7                 ORA     A
0B42 C2470B             JNZ     DBDOS2          ;YES, GO SEND ETX

0B45 3D                 DCR     A               ;NO, INDICATE READY (A == OFFH)
0B46 C9                 RET

0B47 3C         DBDOS2: INR     A               ;FLAG THAT THE NEXT THING TO DO IS WAIT FOR ACK
0B48 12                 STAX    D
```

```
0B49 C5                 PUSH    B

0B4A 0E03               MVI     C,'C' MOD 32    ;SEND THE ETX
0B4C CD650B             CALL    UO

0B4F C1                 POP     B

                BUSY:
0B50 AF         DBDOSB: XRA     A
0B51 C9                 RET

                ;
                ; DUMMY INPUT AND OUTPUT ROUTINES
                ;

0B52 3E1A       DMYIN:  MVI     A,'Z'-40H       ;UNIMPLEMENTED INPUTS RETURN CTL-Z
0B54 C9         DMYOUT: RET                     ;DUMMY OUTPUTS DO NOTHING

                        PAGE
```

```
                        ;
                        ; 8250 I/O ROUTINES
                        ;

                        ; US - GET UART (INPUT) STATUS

0B55 3E05               US:     MVI     A,5             ;OFFSET TO THE STATUS REGISTER
0B57 CD8D0B                     CALL    PINX
0B5A E601                       ANI     1               ;MASK THE DATA AVAILABLE BIT
0B5C C9                         RET

                        ;  UOS - GET UART (OUTPUT) STATUS

0B5D 3E05               UOS:    MVI     A,5             ;OFFSET TO STATUS REG
0B5F CD8D0B                     CALL    PINX
0B62 E620                       ANI     20H
0B64 C9                         RET

                        ; UO - OUTPUT TO UART

0B65 7E                 UO:     MOV     A,M
                        ;       JMP     POUT

                        ;
                        ; POUT - OUTPUT BYTE IN C TO PORT IN A
                        ;

0B66 32740B             POUT:   STA     POUT1+1
0B69 E5                         PUSH    H
0B6A 23                         INX     H
0B6B 23                         INX     H               ;POINT TO FLAG BYTE
0B6C 7E                         MOV     A,M
0B6D 17                         RAL
0B6E E1                         POP     H
0B6F 79                         MOV     A,C
0B70 DC940B                     CC      MUC             ;MAP TO UPPER CASE
0B73 D300               POUT1:  OUT     00H             ;SELF-MODIFYING CODE
0B75 FE0D               POUT2:  CPI     PADCH           ;CHECK IF THIS CHAR NEEDS PADDING (USUALLY CR)
0B77 C0                         RNZ                     ;NO

0B78 E5                         PUSH    H               ;FIND OUT NUMBER OF NULLS REQUIRED
0B79 23                         INX     H
0B7A 23                         INX     H
0B7B 7E                         MOV     A,M             ;GET COUNT FROM DATA STRUCTURE
0B7C E1                         POP     H
0B7D 1F                         RAR                     ;SHIFT INTO LEAST SIG 3 BITS
0B7E 1F                         RAR
0B7F 1F                         RAR
0B80 1F                         RAR
0B81 E607                       ANI     07H             ;MASK ONLY COUNT
0B83 C8                         RZ                      ;RETURN IF NO NULLS ARE REQUIRED
0B84 12                         STAX    D               ;SAVE COUNT OF NULLS TO SEND IN XXXCTS

0B85 C9                         RET
```

```
                        ; UI - INPUT FROM UART

0B86 CD550B     UI:     CALL    US
0B89 CA860B             JZ      UI
0B8C AF         UI1:    XRA     A
                ;       JMP     PINX

                ;
                ; PIN - INPUT BYTE FROM PORT IN A
                ;

0B8D 86         PINX:   ADD     M
0B8E 32920B     PIN:    STA     PIN1+1
0B91 DB00       PIN1:   IN      00H             ;SELF-MODIFYING CODE
0B93 C9                 RET

                ;
                ; MUC - MAP CHARACTER IN A TO UPPER CASE
                ;

0B94 FE61       MUC:    CPI     'a'             ;IF LESS THAN LOWER CASE A
0B96 D8                 RC                      ;  THEN ALREADY UPPER CASE
0B97 FE7B               CPI     'z'+1           ;IF GREATER THAN LOWER CASE Z
0B99 D0                 RNC                     ;  THEN NOT A LOWER CASE LETTER
0B9A D620               SUI     'a'-'A'         ;CONVERT TO UPPER CASE
0B9C C9                 RET

                        PAGE
```

```
                        ; PMSG - PRINT THE MESSAGE AT HL UNTIL NULL
                        ;
0B9D 7E         PMSG:   MOV     A,M             ;GET A CHAR
0B9E B7                 ORA     A               ;CHECK FOR NULL
0B9F C8                 RZ                      ;END OF MESSAGE
0BA0 4F                 MOV     C,A             ; ELSE
0BA1 E5                 PUSH    H               ;SAVE THE POINTER
0BA2 CD0109             CALL    CONOUT          ; PRINT THIS CHARACTER
0BA5 E1                 POP     H
0BA6 23                 INX     H               ; POINT TO NEXT
0BA7 C39D0B             JMP     PMSG            ;REPEAT

                        ;
                        ; HOUT - HEX OUTPUT ROUTINE
                        ;       TYPE CONTENTS OF A IN HEX ON CONSOLE

0BAA F5         HOUT:   PUSH    PSW             ;SAVE CONTENTS OF A
0BAB 0F                 RRC
0BAC 0F                 RRC
0BAD 0F                 RRC
0BAE 0F                 RRC
0BAF CDB30B             CALL    NIBBLE          ;PUT OUT HIGH ORDER NIBBLE
0BB2 F1                 POP     PSW             ;FALL THROUGH TO PUT OUT LOW NIBBLE
0BB3 E60F       NIBBLE: ANI     0FH             ;MASK
0BB5 FE0A               CPI     10              ;> 10 ?
0BB7 FABC0B             JM      NIBBL1          ;IF 0-9
0BBA C607               ADI     7               ;  ELSE CONVERT TO A-F
0BBC C630       NIBBL1: ADI     30H             ;BINARY TO ASCII
0BBE 4F                 MOV     C,A             ;TYPE IT ON THE CONSOLE
0BBF C30109             JMP     CONOUT

                        PAGE
```

```
                        ; BIOS MESSAGES
                        ;
0BC2 0D0A               BTMSG:  DB      CR,LF
0BC4 4552524F52                 DB      'ERROR DURING WARM BOOT - PRESS ANY KEY',0
0BEB 2052454144RDMSG:           DB      ' READ',0
0BF1 2057524954WRMSG:           DB      ' WRITE',0
0BF8 204552524FERRMSG:          DB      ' ERROR ',0
0C00 0D0A00             CRLF:   DB      CR,LF,0

0C03 00                 DCLPOS: DB      0               ;FORCE A CHECK OF LP OUTPUT STATUS
0C04 00                 TTYCTS: DB      0               ;CHRACTERS TO SEND COUNT FOR    TTY
0C05 00                 CRTCTS: DB      0               ;                               CRT
0C06 00                 LPTCTS: DB      0               ;                               LPT
0C07 00                 MDCTS:  DB      0               ;                               MODEM
0C08 00                 DBDCTS: DB      0               ;OUTPUT STATE MACHINE FOR       DBD

                                IF      INTINP
0C09 00                 CRTB:   DB      0               ;NUMBER OF CHARACTERS IN THE CRT BUFFER
0C0A 0D0E               CRTGET: DW      CRTBUF          ;POINTER TO NEXT CHAR TO BE TAKEN
0C0C 0D0E               CRTPUT: DW      CRTBUF          ;POINTER TO NEXT POSITION TO STORE CHAR
                                ENDIF

                                PAGE
```

```
                        IF      H17T
0C0E 1400     DPB17S:   DW      20              ;SEC PER TRACK
0C10 03                 DB      3
0C11 07                 DB      7
0C12 00                 DB      0
0C13 5B00               DW      91              ;DISK SIZE (IN K) - 1
0C15 3F00               DW      63
0C17 C0                 DB      192
0C18 00                 DB      0
0C19 1000               DW      16
0C1B 0300               DW      3
                        ENDIF

                        IF      H47T OR H67T
              DPB0SS:   DW      26              ;SECTORS PER TRACK
                        DB      3,7,0           ;BLOCK SHIFT, BLOCK MASK, EXTENT MASK
                        DW      242             ;DISK SIZE - 1
                        DW      63              ;DIR MAX
                        DB      192,0           ;ALLOC0,ALLOC1
                        DW      16              ;CHECK SIZE
                        DW      2               ;OFFSET
              DPB0SD:   DW      26              ;SECTORS PER TRACK
                        DB      4,15,1          ;BLOCK SHIFT, MASK, EXTENT MASK
                        DW      246             ;DISK SIZE - 1
                        DW      127             ;DIR MAX
                        DB      0C0H,000H       ;ALLOCATION
                        DW      32
                        DW      2

              DPB0DS:   DW      52
                        DB      4,15,0
                        DW      242
                        DW      127
                        DB      0C0H,000H
                        DW      32
                        DW      2

              DPB0DD:   DW      52
                        DB      4,15,0
                        DW      493
                        DW      255
                        DB      0F0H,000H
                        DW      64
                        DW      2
                        ENDIF

                        IF      H47T AND H47ED
              DPB0ES:   DW      64
                        DB      4,15,0
                        DW      299
                        DW      127
                        DB      0C0H,000H
                        DW      32
                        DW      2
```

```
DPBOED: DW      64
        DB      4,15,0
        DW      607
        DW      255
        DB      0F0H,000H
        DW      64
        DW      2
        ENDIF

        PAGE
```

```
                        IF      H17T
0C1D 0102090A11XLT17:   DB      1,2,9,10,17,18
0C23 05060D0E           DB      5,6,13,14
0C27 03040B0C13         DB      3,4,11,12,19,20
0C2D 07080F10           DB      7,8,15,16
                        ENDIF

                        IF      H47T OR H67T
                XLTOS:  DB      1,7,13,19,25
                        DB      5,11,17,23
                        DB      3,9,15,21
                        DB      2,8,14,20,26
                        DB      6,12,18,24
                        DB      4,10,16,22

                XLTOD:  DB      1,2,19,20,37,38
                        DB      3,4,21,22,39,40
                        DB      5,6,23,24,41,42
                        DB      7,8,25,26,43,44
                        DB      9,10,27,28,45,46
                        DB      11,12,29,30,47,48
                        DB      13,14,31,32,49,50
                        DB      15,16,33,34,51,52
                        DB      17,18,35,36
                        ENDIF

                        PAGE
```

```
                        ;
                        ;  HOST BUFFER FLAGS THAT MUST BE INITIALIZED AT ASSEMBLY TIME.
                        ;

0C31 00                 HSTACT  DB      0               ;HOST BUFFER ACTIVE FLAG
0C32 00                 HSTWRT  DB      0               ;HOST BUFFER PENDING WRITE FLAG
0C33 00                 UNACNT  DB      0               ;UNALLOCATED RECORD COUNT


                        ;
0C34 =                  HSTBUF  EQU     $

                        ;
                        ; THE FOLLOWING "ONE-TIME" CODE GETS OVERLAID BY DISK BUFFERS
                        ;  AND POSSIBLY RUN-TIME VARIABLES.
                        ;

                        ;
                        ; BOOT - EXECUTED FOR COLD START
                        ;
0C34 F3                 CBOOT:  DI
0C35 31050F                     LXI     SP,STACK

0C38 3A3400                     LDA     DEFIOB          ;SET THE DEFAULT IOBYTE
0C3B 320300                     STA     IOBYTE

0C3E 3EC3                       MVI     A,MI$JMP
0C40 211B08                     LXI     H,CLOCK         ;ESTABLISH POINTER TO CLOCK INT SERVICE ROUTINE
0C43 320800                     STA     CLKVEC
0C46 220900                     SHLD    CLKVEC+1

                                IF      INTINP
0C49 219B09                     LXI     H,CRTISR        ;POINTER TO CRT INTERRUPT SERVICE ROUTINE
0C4C 321800                     STA     SERVEC          ;AT SERIAL VECTOR
0C4F 221900                     SHLD    SERVEC+1
                                ENDIF

0C52 210D00                     LXI     H,CTLPRT        ;GET THE CURRENT VALUE OF THE RAM AT 0 PORT
0C55 7E                         MOV     A,M             ;  ESTABLISHED BY BLDR
0C56 D3F2                       OUT     H88CTL          ;RESET THE CLOCK ON THE H/Z89

0C58 23                         INX     H               ;POINT TO H8FLAG
0C59 7E                         MOV     A,M
0C5A B7                         ORA     A               ;IF 0 THEN RUNNING ON H/Z89
0C5B CA600C                     JZ      CBT0            ;  THEN DON'T OUTPUT TO 360Q
0C5E D3F0                       OUT     H8CTL           ;  ELSE CONTAINS H8TR TO RESET H8 CLOCK
                        ;
                        ; INITIALIZE 8251 (ONLY IF USED)
                        ;
0C60 3A3600             CBT0:   LDA     MODE            ;FIRST, ASSUME IT IS NOT USED
0C63 E6FE                       ANI     OFFH-MODEBO
0C65 323600                     STA     MODE
```

```
0C68 3A0E00          LDA     H8FLAG          ;IF ON Z/H89
0C6B B7              ORA     A               ; THEN THE CONSOLE IS NOT AN H8-5
0C6C CA9F0C          JZ      CBT1            ;CONSOLE ON H8-4 CARD

0C6F 213700          LXI     H,H84PT1        ;POINT TO CRT PORT
0C72 7E              MOV     A,M             ;GET BASE PORT NUMBER
0C73 C603            ADI     3               ;SEE IF YOU CAN GET A RESPONSE FROM 8250 @ 350Q
0C75 32A50E          STA     OUTH84+1
0C78 F5              PUSH    PSW
0C79 3E03            MVI     A,3             ;SET 8 BIT WORDS
0C7B CDA40E          CALL    OUTH84
0C7E F1              POP     PSW
0C7F CD8E0B          CALL    PIN
0C82 FE03            CPI     3
0C84 CA9F0C          JZ      CBT1            ;CONSOLE ON H8-4 CARD
0C87 3A3600          LDA     MODE
0C8A F601            ORI     MODEB0          ;CONSOLE MUST BE ON H8-5 CARD THEN
0C8C 323600          STA     MODE            ; SO SET MODE

0C8F 3E15            MVI     A,15H           ;DUMMY MODE BYTE
0C91 D3FB            OUT     H85CRT+1

0C93 3E40            MVI     A,40H           ;RESET 8251
0C95 D3FB            OUT     H85CRT+1

0C97 3E4E            MVI     A,4EH           ;8 BIT WORDS, 1 STOP BIT, NO PARITY
0C99 D3FB            OUT     H85CRT+1

                     IF      NOT INTINP
                     MVI     A,15H           ;ENABLE TX AND RX WITH INTERRUPTS OFF
                     ENDIF
                     IF      INTINP
0C9B 3E17            MVI     A,17H           ;ENABLE TX AND RX WITH INTERRUPTS ON
                     ENDIF
0C9D D3FB            OUT     H85CRT+1

                ;
                ; NOW INITIALIZE THE 8250S
                ;
0C9F 2A3800     CBT1:LHLD    CRTBAUD         ;PICK UP BAUD RATE
0CA2 3A3700          LDA     H84PT1          ;  AND THE PORT NUMBER
0CA5 CD640E          CALL    IN8250          ;INITIALIZE THIS UART
                ;
                     IF      INTINP
0CA8 3A3700          LDA     H84PT1
0CAB 3C              INR     A               ;POINT TO INTERRUPT ENABLE REGISTER
0CAC 32A50E          STA     OUTH84+1
0CAF 3E01            MVI     A,1             ;ENABLE RECEIVER INTERRUPTS
0CB1 CDA40E          CALL    OUTH84
                     ENDIF

0CB4 2A3B00          LHLD    TTYBAUD
0CB7 3A3A00          LDA     H84PT2
0CBA CD640E          CALL    IN8250
                ;
0CBD 2A3E00          LHLD    LPTBAUD
```

```
0CC0 3A3D00          LDA     H84PT3
0CC3 CD640E          CALL    IN8250
                ;
0CC6 2A4100          LHLD    RDPBAUD
0CC9 3A4000          LDA     H84PT4
0CCC CD640E          CALL    IN8250

                ;  PRINT SIGNON MESSAGE.

0CCF 21340D          LXI     H,SMSG0         ;PRINT THE SIGNON MESSAGE
0CD2 CD9D0B          CALL    PMSG
0CD5 3E10            MVI     A,(BIOSEND+255)/256 ;FINE OUT NEWLY RELOCATED SIZE
0CD7 1F              RAR                     ;GET THE VALUE IN K BY DIVIDING BY 4
0CD8 1F              RAR                     ; (DONE AT RUN TIME FOR RELOCATION)
0CD9 E63F            ANI     03FH
0CDB C2E00C          JNZ     CBOOT1          ;IF THE TOP OF MEMORY IS NOT 0000H
0CDE 3E40            MVI     A,64            ; ELSE TAKE CARE OF THE 64K CASE
0CE0 CD480E    CBOOT1: CALL  TYDN            ;TYPE A 2 DIGIT DECIMAL NUMBER
0CE3 21380D          LXI     H,SMSG1
0CE6 CD9D0B          CALL    PMSG

                ;
                ;  DO INITIALIZATION FOR BOOT DEVICE.
                ;

0CE9 3A4900    CBI:  LDA     BBDA            ;GET BOOT DEVICE BASE PORT #
0CEC 32800D          STA     CBIB

0CEF 21070D          LXI     H,CBI1          ;SET RETURN ADDR
0CF2 E5              PUSH    H

0CF3 3A0400          LDA     LOGDSK          ;GET BOOT UNIT #
0CF6 4F              MOV     C,A

0CF7 3A4800          LDA     BBDF            ;GET BOOT DEVICE FLAGS
0CFA E6E0            ANI     DPETYPE

                     IF      H17T
0CFC FE40            CPI     DPEH17
0CFE CA820D          JZ      CBH17           ;BR IF H17
                     ENDIF

                     IF      H37T
                     CPI     DPEH37
                     JZ      CBH37           ;BR IF H37
                     ENDIF

                     IF      H47T
                     CPI     DPEH47
                     JZ      CBH47           ;BR IF H47
                     ENDIF

                     IF      H67T
                     CPI     DPEH67H
                     JZ      CBH67           ;BR IF H67
                     ENDIF
```

```
0D01 0E07              MVI     C,BELL          ;BIOS IS NOT SET UP TO HANDLE
0D03 CD0109            CALL    CONOUT          ; BOOT DEVICE
0D06 76                HLT                     ;HALT EVERYTHING

                ;
                ;  DO INITITIALIZATION FOR OTHER DISK DEVICE (IF ANY).
                ;

0D07 AF         CBI1:  XRA     A               ;INDICATE NOT BOOT DEVICE
0D08 327F0D            STA     CBIA

0D0B 3A800D            LDA     CBIB            ;SWITCH BASE PORT #
0D0E EE04              XRI     7CH-78H         ;DRIVE BASE PORT #'S ARE 78H & 7CH
0D10 32800D            STA     CBIB            ; WHICH DIFFER ONLY AT BIT 2

0D13 21240D            LXI     H,CBI2          ;SET RET ADDR
0D16 E5                PUSH    H

0D17 3A4800            LDA     BBDF            ;GET DEVICE FLAGS OF BOOT DEVICE
0D1A E6E0              ANI     DPETYPE         ;MASK FOR DRIVE TYPE
0D1C 0E00              MVI     C,0             ;1ST UNIT IS UNIT 0

                       IF      H17T
0D1E FE40              CPI     DPEH17          ;BOOT DEVICE H17
0D20 C2820D            JNZ     CBH17           ;IF NOT, THEN OTHER DEVICE IS H17
                       ENDIF

                       IF      H37T
                       CPI     DPEH37          ;BOOT DEVICE H37
                       JNZ     CBH37           ;IF NOT, THEN OTHER DEVICE IS H37
                       ENDIF

                       IF      H47T
                       CPI     DPEH47          ;BOOT DEVICE H47
                       JNZ     CBH47           ;IF NOT, THEN OTHER DEVICE IS H47
                       ENDIF

                       IF      H67T
                       CPI     DPEH67H         ;BOOT DEVICE H67
                       JNZ     CBH67           ;IF NOT, THEN OTHER DEIVCE IS H67
                       ENDIF

0D23 E1                POP     H               ;NO OTHER DEVICE, DISCARD RET ADDR

                CBI2:

                ;
                ;  FINISH COLD BOOT AND SIGN ON.
                ;

0D24 AF                XRA     A               ;MAKE A THE DEFAULT DRIVE
0D25 320400            STA     LOGDSK

0D28 210000            LXI     H,BIOS          ;PLACE ADDRESS OF START OF BIOS
0D2B 224E00            SHLD    BBIOS           ; IN PAGE ZERO
```

```
0D2E FB                 EI

0D2F 3E00               MVI     A,BT$CD         ;FLAG AS A COLD BOOT
0D31 C32A01             JMP     GOW

0D34 0D0A0A00  SMSG0:   DB      CR,LF,LF,0
0D38 4B2048454 1SMSG1:  DB      'K HEATH/ZENITH CP/M 2.2X'
0D50 303320             DB      VERS/10+'0',(VERS MOD 10)+'0',LEVEL
0D53 2030382F32         DB      ' ',MONTH/10+'0',(MONTH MOD 10)+'0','/',DAY/10+'0'
0D58 382F3831           DB      (DAY MOD 10)+'0','/',YEAR/10+'0',(YEAR MOD 10)+'0'
0D5C 0D0A               DB      CR,LF
0D5E 464F52             DB      'FOR'
                        IF      H17T
0D61 20483137           DB      ' H17'
                        ENDIF
                        IF      H37T
                        DB      ' H37'
                        ENDIF
                        IF      H47T
                        DB      ' H47'
                        ENDIF
                        IF      H67T
                        DB      ' H67'
                        ENDIF
0D65 204449534B         DB      ' DISKS'
0001 =          WRKVARX EQU     PARTITN OR H67PART2 OR INTINP OR BRKKEY OR TOD OR EVENT
0001 =          WRKVAR  EQU     WRKVARX OR H37ED OR H47ED
                        IF      WRKVAR
0D6B 2057495448         DB      ' WITH OPTION(S) '
                        IF      PARTITN
                        DB      'P'
                        ENDIF
                        IF      H67PART2
                        DB      '2'
                        ENDIF
                        IF      TOD
                        DB      'T'
                        ENDIF
                        IF      EVENT
                        DB      'E'
                        ENDIF
                        IF      INTINP
0D7B 49                 DB      'I'
                        ENDIF
                        IF      BRKKEY
                        DB      'B'
                        ENDIF
                        IF      H37ED
                        DB      'E3'
                        ENDIF
                        IF      H47ED
                        DB      'E4'
                        ENDIF
                        ENDIF
0D7C 0D0A00             DB      CR,LF,0
```

PAGE

```
0D7F 01        CBIA:   DB      1               ;0 = INIT AS OTHER DEVICE
                                               ;1 = INIT AS BOOT DEVICE
0D80           CBIB:   DS      1               ;BASE PORT #
0D81 00        CBIC:   DB      0               ;LOGICAL DRIVE #

                       IF      H17T
                ;
                ;  COLD BOOT INIT ROUTINE FOR H17.
                ;

0D82 3A800D    CBH17:  LDA     CBIB            ;CHECK IF VALID BASE PORT #
0D85 FE7C              CPI     7CH
0D87 C0                RNZ                     ;ONLY 7CH ALLOWED

0D88 0603              MVI     B,H17ND
0D8A 1600              MVI     D,(DPE0-DPBASE)/DPEL
0D8C CD9C0D            CALL    CBTFIL          ;FILL IN DRIVE MAP TABLE

0D8F 3E10              MVI     A,DFMO          ;INIT H17 CONTROL REG IMAGE
0D91 320F00            STA     DEVCTL

0D94 3A7F0D            LDA     CBIA            ;CHECK IF BOOT DEVICE
0D97 A7                ANA     A
0D98 CC2906            CZ      RESH17          ;IF NOT, THEN RESET DEVICE

0D9B C9                RET
                       ENDIF

                       IF      H37T
                ;
                ;  H37 COLB BOOT INIT ROUTINE.
                ;

                CBH37:  LDA     CBIB            ;CHECK IF VALID BASE PORT #
                        CPI     78H
                        RNZ                     ;ONLY 78H ALLOWED

                        MVI     B,H37ND
                        MVI     D,(DPE37$0-DPBASE)/DPEL
                        CALL    CBTFIL          ;FILL IN DRIVE MAP TABLE

                        MVI     A,MI$JMP        ;INSTALL H37 INTERRUPT SERVICE ROUTINE
                        LXI     H,H37ISR
                        STA     H37VEC
                        SHLD    H37VEC+1

                        LDA     CBIA            ;CHECK IF BOOT DEVICE
                        ANA     A
                        JZ      RESH37          ;IF NOT, RETURN THRU RESET DEVICE

                        LXI     H,DLYMO37       ;INIT VALUES
                        XRA     A
                        MOV     M,A
                        INX     H
                        MOV     M,A
```

```
                MVI     A,CONMO         ;INIT CONTROL REG IMAGE
                STA     H37CTL

                RET
                ENDIF

                IF      H47T
                ;
                ;  H47 COLD BOOT INIT ROUTINE.
                ;

        CBH47:  MVI     B,H47ND
                MVI     D,(DPE47$0-DPBASE)/DPEL
                CALL    CBTFIL          ;FILL IN DRIVE MAP TABLE

                LDA     CBIB            ;SET PORT #'S
                IF      H47CTL
        %:      H47CTL NE 0
                ENDIF
                STA     H47INS1         ;STATUS PORT #
                STA     H47OUTC1        ;CONTROL PORT #
                INR     A
                IF      H47CTL+1-H47DAT
        %:      H47DAT NE H47CTL+1
                ENDIF
                STA     H47IND1         ;INPUT DATA PORT #
                STA     H47OUTD1        ;OUTPUT DATA PORT #

                LDA     CBIB            ;CHECK BASE PORT ADDR
                CPI     7CH
                JNZ     CBH472          ; BR IF NOT PORT 7CH
                MVI     B,H47ND
                LXI     D,DPEL
                LXI     H,DPE47$0+DPEFLAG
        CBH471: MOV     A,M
                ORI     DPEP7C          ;SET PORT 7CH FLAG
                MOV     M,A
                DAD     D
                DCR     B
                JNZ     CBH471

        CBH472: LDA     CBIA            ;CHECK IF BOOT DEVICE
                ANA     A
                CZ      RESH47          ;IF NOT, THEN RESET DEVICE

                RET
                ENDIF

                IF      H67T
                ;
                ;  H67 COLD BOOT INIT ROUTINE.
                ;

        CBH67:

                IF      H67PART2
```

```
        LDA     BBDF            ;Q. BOOTED FROM FLOPPY
        ANI     DPETYPF
        CPI     DPEH67F
        JNZ     CBH670          ; BR IF NOT
        INR     C               ;BUMP LOGICAL UNIT #
CBH670:
        ENDIF

        MVI     B,H67ND
        MVI     D,(DPE67$0-DPBASE)/DPEL
        CALL    CBTFIL          ;FILL IN DRIVE MAP TABLE

        LDA     CBIB            ;SET PORT #'S
        IF      HD$DAT
%:      HD$DAT NE 0
        ENDIF
        STA     H67IND1         ;INPUT DATA PORT #
        STA     H67OUTD1        ;OUTPUT DATA PORT #
        INR     A
        IF      (HD$DAT+1-HD$STA) OR (HD$DAT+1-HD$CON)
%:      (HD$STA NE (HD$DAT+1)) OR (HD$CON NE (HD$DAT+1))
        ENDIF
        STA     H67INS1         ;STATUS PORT #
        STA     H67OUTC1        ;CONTROL PORT #

        LDA     CBIB            ;CHECK BASE PORT ADDR
        CPI     7CH
        JNZ     CBH672          ; BR IF NOT PORT 7CH
        MVI     B,H67ND
        LXI     D,DPEL
        LXI     H,DPE67$0+DPEFLAG
CBH671: MOV     A,M
        ORI     DPEP7C          ;SET PORT 7CH FLAG
        MOV     M,A
        DAD     D
        DCR     B
        JNZ     CBH671

CBH672: LDA     CBIA            ;CHECK IF BOOT DEVICE
        ANA     A
        JZ      RESH67          ;IF NOT, THEN RESET DEVICE & RET

        IF      PARTITN
        LDA     BBDF            ;GET BOOT DEVICE FLAGS
        ANI     DPETYPF
        CPI     DPEH67H         ;CHECK FOR HARD DISK
        RNZ                     ;RET IF NOT HARD DISK

        LDA     DPE67$0+DPEFLAG ;MARK PARTITION IS ASSIGNED
        ORI     DPEASGN
        STA     DPE67$0+DPEFLAG
        LHLD    BBP             ;GET SECTOR # FOR BEGINNING
        SHLD    DPE67$0+DPETRK  ; OF PARTITION
        LHLD    BUPB            ;GET LAST SECTOR # + 1
        SHLD    DPE67$0+DPEUPB  ; OR PARTITION
        ENDIF
```

```
RET
ENDIF

PAGE
```

```
                        ; CBTFIL - FILL THE LOGICAL TO PHYSICAL MAPPING TABLE
                        ;          FOR REAL DRIVES.  THEN DO THE SAME FOR THE IMAGINARY
                        ;          DRIVES PLUS SET UP THE IMAGINARY'S LINK TO HIS
                        ;          CORRESPONDING REAL DRIVE.
                        ;
                        ;    UPON ENTRY -- (B) = # DRIVES
                        ;                  (C) = STARTING UNIT #
                        ;                  (D) = STARTING DRIVE MAP #
                        ;                  (CBIC) = NEXT LOGICAL DRIVE # TO BE ASSIGNED
                        ;    UPON EXIT  -- (CBIC) UPDATED
                        ;    USES       -- ALL
                        ;

                        CBTFIL:
0D9C 210000                     LXI     H,0             ;INIT CBTIA
0D9F 22460E                     SHLD    CBTIA

                        CBTFIL1:
0DA2 C5                         PUSH    B
0DA3 D5                         PUSH    D

0DA4 3A7F0D                     LDA     CBIA            ;IF BOOT DEVICE THEN INSURE
0DA7 A7                         ANA     A               ; BOOT UNIT IS MARKED REAL
0DA8 C4300E                     CNZ     CBTF7

0DAB AF                         XRA     A               ;INDICATE REAL CYCLE
0DAC CDC10D                     CALL    CBTF0           ;HANDLE REAL DRIVES

0DAF D1                         POP     D
0DB0 C1                         POP     B

0DB1 2A460E                     LHLD    CBTIA           ;IF NO REAL DRIVES THEN MARK
0DB4 7C                         MOV     A,H             ; 1ST AVAILABLE DRIVE AS REAL
0DB5 B5                         ORA     L
0DB6 C2BF0D                     JNZ     CBTFIL2         ;  BR IF REAL DRIVE FOUND

0DB9 CD300E                     CALL    CBTF7           ;MARK 1ST AVAILABLE DRIVE REAL

0DBC C3A20D                     JMP     CBTFIL1         ;GO THRU REAL DRIVE CYCLE AGAIN

                        CBTFIL2:
0DBF 3E01                       MVI     A,1             ;INDICATE IMAGINARY CYCLE

                        ;  THIS SECTION OF CODE IS RAN THROUGH TWICE.
                        ;  FIRST TIME IS FOR HANDLING THE REAL DRIVES.
                        ;  SECOND TIME IS FOR HANDLING THE IMAGINARY DRIVES.

0DC1 32450E             CBTF0:  STA     CBTFA           ;SAVE TYPE OF CYCLE INDICATOR

0DC4 58                         MOV     E,B             ;COPY OF # DRIVES

0DC5 79                 CBTF1:  MOV     A,C             ;GET THIS DRIVE'S NUMBER
0DC6 B8                         CMP     B               ;MOD # DRIVES
0DC7 DACB0D                     JC      CBTF2
0DCA 90                         SUB     B
```

```
0DCB 82        CBTF2:  ADD     D               ;COMPUTE DISK ENTRY TABLE TO USE

0DCC C5                PUSH    B
0DCD 47                MOV     B,A             ;COMBINE LOGICAL AND MAPPED DRIVE #'S
0DCE 3A810D            LDA     CBIC            ; BIT 7-4 = LOGICAL
0DD1 87                ADD     A               ; BIT 3-0 = MAPPED
0DD2 87                ADD     A
0DD3 87                ADD     A
0DD4 87                ADD     A
0DD5 80                ADD     B
0DD6 47                MOV     B,A             ;(B) = LOGICAL/MAPPED DRIVE #'S
0DD7 D5                PUSH    D
0DD8 CDAB08            CALL    GETDPEX         ;GET ADDR OF DPE'S HEATH EXTENSIONS
0DDB EB                XCHG                    ;(DE) = HEATH EXTENSIONS
0DDC 210600            LXI     H,DPEFLG2-DPEHTH
0DDF 19                DAD     D
0DE0 7E                MOV     A,M
0DE1 E602              ANI     DPEIMG          ;CHECK IF IMAGINARY DRIVE
0DE3 3A450E            LDA     CBTFA           ;GET CYCLE INDICATOR ALSO
0DE6 C2040E            JNZ     CBTF3           ;BR IF IMAGINARY DRIVE

                ;  HANDLE REAL DRIVE IF THIS IS REAL DRIVE CYCLE.
0DE9 A7                ANA     A
0DEA C2190E            JNZ     CBTF4           ;BR IF THIS IS NOT REAL DRIVE CYCLE

0DED E5                PUSH    H               ;IF THIS IS FIRST REAL DRIVE ENCOUNTERED
0DEE 2A460E            LHLD    CBTIA           ; THEN REMEMBER ITS HEATH EXTENSION
0DF1 7C                MOV     A,H             ;  ADDRESS
0DF2 B5                ORA     L
0DF3 E1                POP     H
0DF4 C2FC0D            JNZ     CBTF2A
0DF7 EB                XCHG
0DF8 22460E            SHLD    CBTIA
0DFB EB                XCHG

0DFC 23        CBTF2A: INX     H               ;PLACE LOGICAL/MAPPED IN DPELUN SLOT
                       IF      DPEFLG2+1-DPELUN
                %:     (DPEFLG2+1) NE DPELUN
                       ENDIF
0DFD 70                MOV     M,B

0DFE CD210E            CALL    CBTF6           ;MAP DRIVE

0E01 C3190E            JMP     CBTF4

                ;  HANDLE IMAGINARY DRIVE IF THIS IS THE IMAGINARY DRIVE CYCLE.
0E04 A7        CBTF3:  ANA     A
0E05 CA190E            JZ      CBTF4           ;BR IF THIS IS NOT IMAGINARY DRIVE CYCLE

0E08 E5                PUSH    H               ;(DE) = ADDR OF HEATH EXTENSIONS
                                               ;(HL) = ADDR OF 2ND FLAG BYTE

0E09 CD210E            CALL    CBTF6           ;MAP DRIVE

0E0C 2A460E            LHLD    CBTIA           ;MOVE THE REAL DRIVE'S TABLE
0E0F 0E08              MVI     C,DPEHL         ; INTO THIS IMAGINARY DRIVE'S TABLE
```

```
0E11 CDB908            CALL    MOVEIT

0E14 E1                POP     H               ;(HL) = ADDR OF 2ND FLAG BYTE
0E15 7E                MOV     A,M
0E16 F602              ORI     DPEIMG          ;REMARK AS IMAGINARY DRIVE
0E18 77                MOV     M,A

0E19 D1        CBTF4:  POP     D
0E1A C1                POP     B
0E1B 0C                INR     C               ;ROUND ROBIN TO NEXT DRIVE
0E1C 1D                DCR     E               ;COUNT THIS ONE AS DONE
0E1D C2C50D            JNZ     CBTF1

0E20 C9                RET

               ;  PLACE MAPPED DRIVE # INTO MAP DRIVE TABLE.
               ;  (B) = MAPPED DRIVE #

0E21 3A810D    CBTF6:  LDA     CBIC            ;GET LOGICAL DRIVE #
0E24 214000            LXI     H,BDMAP         ;GET ADDR OF MAP DRIVE
0E27 CD9508            CALL    DADA            ; TABLE SLOT
0E2A 70                MOV     M,B             ;PLACE MAPPED DRIVE # THERE
0E2B 21810D            LXI     H,CBIC          ;BUMP VALUE NEXT LOGICAL DRIVE #
0E2E 34                INR     M
0E2F C9                RET

               ;  INSURE 1ST DRIVE OF GROUP IS REAL DRIVE (NOT IMAGINARY).
               ;  THE 1ST DRIVEHAS TO BE MARKED REAL SO IMAGINARY DRIVES,
               ;  IF ANY, HAVE A REAL UNIT TO USE.

               CBTF7:
0E30 C5                PUSH    B
0E31 D5                PUSH    D

0E32 79                MOV     A,C             ;GET ADDR OF HEATH EXTENSIONS
0E33 82                ADD     D
0E34 CDAB08            CALL    GETDPEX
0E37 22460E            SHLD    CBTIA           ;SAVE ADDR OF ITS DPE'S HEATH
                                               ; EXTENSIONS FOR IMAGINARY DRIVES
0E3A 110600            LXI     D,DPEFLG2-DPEHTH
0E3D 19                DAD     D
0E3E 7E                MOV     A,M
0E3F E6FD              ANI     OFFH-DPEIMG     ;MARK REAL (NOT IMAGINARY)
0E41 77                MOV     M,A

0E42 D1                POP     D
0E43 C1                POP     B

0E44 C9                RET

0E45           CBTFA   DS      1               ;CYCLE TYPE INDICATOR SLOT
0E46           CBTIA   DS      2               ;ADDR OF REAL DRIVE'S DPE HEATH EXT.

                       PAGE
```

```
                        ; TYDN - TYPE A TWO DIGIT DECIMAL NUMBER ON CONSOLE
                        ;        ENTRY   A       VALUE
                        ;

0E48 0E00       TYDN:   MVI     C,0             ;INITIALIZE QUOTIENT
0E4A D60A       TYDN1:  SUI     10              ;REPEATEDLY SUBTRACT 10
0E4C DA530E             JC      TYDN2           ;IF UNDERFLOW
0E4F 0C                 INR     C               ;ELSE INCREMENT THE QUOTIENT
0E50 C34A0E             JMP     TYDN1           ;AND SUBTRACT AGAIN

0E53 C60A       TYDN2:  ADI     10              ;CORRECT THE UNDERFLOW
0E55 F5                 PUSH    PSW             ;SAVE THE REMAINDER
0E56 79                 MOV     A,C             ;GET THE QUOTIENT
0E57 C630               ADI     030H            ;ASCII ADJUST IT
0E59 4F                 MOV     C,A
0E5A CD0109             CALL    CONOUT          ;SEND IT TO CONSOLE
0E5D F1                 POP     PSW             ;RECALL REMAINDER
0E5E C630               ADI     030H            ;ASCII ADJUST
0E60 4F                 MOV     C,A
0E61 C30109             JMP     CONOUT          ;PRINT IT, WITH IMPLICIT RETURN

                        PAGE
```

```
                        ; IN8250 - INITIALIZE AN 8250
                        ;    HL CONTAINS BAUD RATE DIVISOR (WORD)
                        ;    A  HAS BASE PORT NUMBER
                        ;
0E64 47         IN8250: MOV     B,A             ;SAVE BASE PORT NUMBER IN B
0E65 EB                 XCHG                    ;MOVE BAUD RATE DIVISOR TO DE
0E66 21A50E             LXI     H,OUTH84+1      ;POINT TO PORT IN OUT INSTRUCTION
0E69 3E03               MVI     A,3             ;BAUD RATE ACCESS BIT ON BASE+3 PORT
0E6B 80                 ADD     B               ;GET ACTUAL PORT
0E6C 4F                 MOV     C,A             ;SAVE IN C FOR LATER
0E6D 77                 MOV     M,A             ;AND MODIFY OUTPUT INSTRUCTION
0E6E 3E83               MVI     A,83H           ;SET DIVISOR LATCH ACCESS BIT
0E70 CDA40E             CALL    OUTH84          ;TO A "1"
0E73 34                 INR     M               ;POINT TO MODEM CONTROL REGISTER
0E74 3E0F               MVI     A,0FH           ;AND SET DSR & CTS HIGH FOR DIABLO
0E76 CDA40E             CALL    OUTH84          ; AND OTHER TERMINALS WHICH REQUIRE THEM
0E79 70                 MOV     M,B             ;SET PORT TO LEAST SIG BYTE
0E7A 7B                 MOV     A,E
0E7B CDA40E             CALL    OUTH84
0E7E 7A                 MOV     A,D             ;NOW DO MOST SIG BYTE
0E7F E60F               ANI     0FH             ;AND OFF CONTROL FLAGS
0E81 34                 INR     M               ;ON NEXT PORT
0E82 CDA40E             CALL    OUTH84
0E85 71                 MOV     M,C             ;RESET PORT TO DIVISOR LATCH ACCESS
0E86 FE04               CPI     B110 SHR 8      ;IF SET FOR GREATER THAN 110
0E88 3E03               MVI     A,3             ; THEN SET NO PARITY, 8 BIT WORDS, 1 STOP BIT
0E8A DA8F0E             JC      IN821
0E8D F604               ORI     4               ; ELSE SET TWO STOP BITS FOR 110 AND BELOW
0E8F CDA40E     IN821:  CALL    OUTH84
0E92 35                 DCR     M               ;NOW SET PORT FOR INTERRUPT CONTROL
0E93 35                 DCR     M
0E94 AF                 XRA     A               ;DISABLE ALL DEVICE INTERRUPTS
0E95 CDA40E             CALL    OUTH84          ;DISABLE INTS

                        ; DELAY FOR APPROXIMATELY TWO CHARACTER TIMES

0E98 EB                 XCHG                    ;PUT BAUD RATE DIVISOR IN HL
0E99 29                 DAD     H               ;MULTIPLY BY 16 TO GET DELAY
0E9A 29                 DAD     H
0E9B 29                 DAD     H
0E9C 29                 DAD     H
0E9D 2B         LOOP1:  DCX     H
0E9E 7D                 MOV     A,L
0E9F B4                 ORA     H
0EA0 C29D0E             JNZ     LOOP1
0EA3 C9                 RET

                        ; SELF MODIFYING OUT INSTRUCTION USED BY IN8250

0EA4 D300       OUTH84: OUT     0               ;PORT IS MODIFIED
0EA6 C9                 RET

0273 =          CLEN    EQU     $-HSTBUF        ;COLD BOOT CODE LENGTH
```

```
                        ;+*******************
                        ;+
                        ;+  IF COLD BOOT CODE IS SMALLER THAN HOST BUFFER,
                        ;+  THEN FILL OUT 'HSTBUF' WITH DS STATEMENT.
                                IF      (CLEN-HSTSIZ) SHR 15
                                DS      HSTSIZ-CLEN
                        ;+  OTHERWISE REORG SO RUN-TIME VARIABLES CAN ALSO OVERLAY
                        ;+  COLD BOOT CODE.
                                ELSE
0D34                            ORG     HSTBUF+HSTSIZ
                                ENDIF
                        ;+
                        ;+*******************

0D34                    DIRBUF: DS      128

                                IF      H17T
0DB4                    ALV0:   DS      12
0DC0                    CSV0:   DS      16
0DD0                    ALV1:   DS      12
0DDC                    CSV1:   DS      16
0DEC                    ALV2:   DS      12
0DF8                    CSV2:   DS      16
                                ENDIF

                                IF      H37T
                        DPB37$0 DS      DPBL
                        ALV37$0 DS      50
                        CSV37$0 DS      64
                        DPB37$1 DS      DPBL
                        ALV37$1 DS      50
                        CSV37$1 DS      64
                        DPB37$2 DS      DPBL
                        ALV37$2 DS      50
                        CSV37$2 DS      64
                                ENDIF

                                IF      H47T
                        ALV47$0 DS      77
                        CSV47$0 DS      64
                        ALV47$1 DS      77
                        CSV47$1 DS      64
                                ENDIF

                                IF      H67T
                        DPB67$0 DS      DPBL
                        ALV67$0 DS      256
                        ALV67$1 DS      77
                        CSV67$1 DS      64
                                IF      H67PART2
                        DPB67$2 DS      DPBL
                        ALV67$2 DS      256
                                ENDIF
                                ENDIF
```

```
0E08          DPBX:   DS      2
0E0A          HSTDPB: DS      2

0E0C          DMAB:   DS      2               ;DMA BUFFER - USED TO STORE STARTING ADDRESS
                                              ;  OF TRACK DURING WARM BOOT
0E0E          SPT:    DS      1               ;NUMBER OF SECTORS PER TRACK (DURING WBOOT)
0E0F          XLTW:   DS      2               ;SECTOR XLATE TABLE (DURING WBOOT)
0E11          SPT1:   DS      1               ;DITTO, BEYOND TRACK 0
0E12          XLTW1:  DS      2               ;DITTO, BEYOND TRACK 0

0E14          TRACK:  DS      1               ;       TRACK (# < 256)
0E15          SECTOR: DS      1               ;       SECTOR
0E16          SIDE:   DS      1
0E17          RWOP    DS      1               ;I/O TYPE OPERATION
                                              ;0=READ  1=WRITE
0E18          LSP:    DS      1               ;LOGICAL SECTORS PER PHYSICAL
0E19          ERRCNT: DS      1               ;RETRY COUNTER
0E1A          ERRTYP: DS      1               ;TYPE OF ERROR
0E1B          TRKPT:  DS      2               ;CONTAINS POINTER TO TRACK REGISTER
                                              ;FOR CURRENT DRIVE
              ;
0E1D          SEKDSK: DS      1               ;SEEK DISK NUMBER
0E1E          SEKTRK: DS      2               ;SEEK TRACK NUMBER
0E20          SEKSEC: DS      1               ;SEEK SECTOR NUMBER
              ;
0E21          HSTDSK: DS      1               ;HOST DISK NUMBER
0E22          HSTTRK: DS      2               ;HOST TRACK NUMBER
0E24          HSTSEC: DS      1               ;HOST SECTOR NUMBER
              ;
0E25          SEKHST: DS      1               ;SEEK SHR SECSHF
              ;
0E26          UNADSK: DS      1               ;LAST UNALLOC DISK
0E27          UNATRK: DS      2               ;LAST UNALLOC TRACK
0E29          UNASI:  DS      1               ;LAST UNALLOC SECTOR INDEX INTO XLT TABLE
              ;
0E2A          ERFLAG: DS      1               ;ERROR REPORTING
0E2B          RSFLAG: DS      1               ;READ SECTOR FLAG
0E2C          READOP: DS      1               ;1 IF READ OPERATION
0E2D          WRTYPE: DS      1               ;WRITE OPERATION TYPE
0E2E          DMAADR: DS      2               ;LAST DMA ADDRESS

                      PAGE
```

```
                        ;********************
                        ;
                        ;  THE FOLLOWING AREAS CANNOT OVERLAY THE COLD BOOT CODE, SINCE
                        ;  THEY CAN/ARE USED DURING COLD BOOT.
                        ;
                        ;  IF HOST BUFFER + RUN-TIME VARIABLES OCCUPY LESS SPACE THAN THE
                        ;  COLD BOOT CODE, IT IS NECESSARY TO REORG PAST THE COLD BOOT CODE.
                        ;
                        ;********************

                                IF      (($-HSTBUF)-CLEN) SHR 15
0EA7                            ORG     HSTBUF+CLEN
                                ENDIF

0EA7                    HSAV:   DS      2               ;SAVED HL DURING INTERRUPT SERVICE
0EA9                    RETSAV: DS      2               ;SAVED RETURN ADDRESS DURING INT SERVICE
0EAB                    OLDSP:  DS      2               ;SAVE SP DURING INTERRUPT SERVICE

                                IF      INTINP

0EAD                    CRTBUF: DS      40              ;CRT TYPE-AHEAD BUFFER
0ED5 =                  CRTBND  EQU     $
0028 =                  CRTLEN  EQU     CRTBND-CRTBUF

0ED5                            DS      16              ;LOCAL STACK FOR CRT INTERRUPT SERVICE
0EE5 =                  LCLSTK  EQU     $

                                ENDIF

0EE5                            DS      32              ;STACK AREA DURING COLD & WARM BOOT
0F05                    STACK   DS      0

0F05 =                  BIOSEND EQU     $

0F05                            END
```